

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Conception et développement d'un logiciel graphique interactif destiné à la résolution de problèmes scientifiques

Van Wonterghem, Vincent

Award date:
1983

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

CONCEPTION ET DEVELOPPEMENT
D'UN LOGICIEL GRAPHIQUE INTERACTIF
DESTINE A
LA RESOLUTION DE PROBLEMES SCIENTIFIQUES.

Mémoire présenté pour l'obtention
du grade de "Licencié et Maître
en Informatique"

par

Vincent VAN WONTERGHEM

- 1983 -

*FM
B16/1983/
02

BUMP

J'exprime ma gratitude envers Messieurs les Professeurs J.Ramaekers et J.Delhalle pour m'avoir permis de mener ce travail à bien dans les meilleures conditions.

Mes remerciements vont aussi au Docteur J.Lejeune pour ses explications concernant le logiciel graphique WAND.

Je tiens aussi à remercier Mme A.Dierge pour sa patience et ses nombreux conseils au sujet de l'utilisation de l'ordinateur PDP 11/60.

Enfin, je remercie ma gentille dactylo pour le tracé des figures, la mise en page et la dactylographie du texte.

TABLE DES MATIERES.

- I. Introduction générale.
- II. Introduction aux problèmes graphiques à l'origine de ce travail.
- III. Etude de la configuration et de l'architecture du système graphique.
 - 1) Configuration
 - 2) Architecture du Megatek
 - 3) Exécution d'un programme graphique
- IV. Analyse d'opportunité.
 - 1) Analyse de l'existant
 - Description de la chaîne de programmes
 - a) Conventions utilisées dans les schémas
 - b) Schéma de la chaîne de programmes
 - c) Description des programmes graphiques
 - 2) Critique de l'existant
 - 3) Expression des besoins
 - a) Objectifs généraux
 - b) Ebauche de solutions
 - Pour la représentation des molécules
 - Pour l'ordonnancement des bandes
 - Pour la représentation à la table traçante
- V. Proposition de standardisation des problèmes graphiques.
 - 1) Principe
 - 2) Moyen
 - a) Informations nécessaires pour la description d'un élément du multigraphe (sommet ou arête)

- b) Numérotation utilisateur et numérotation système
- c) Choix d'une structure de données

3) Librairie GRALIB

- a) Fonctionnement général de la librairie
- b) Règles d'utilisation
- c) Description des différentes routines existantes

4) Contraintes inhérentes à la solution implémentée

VI. Description générale des solutions.

1) Point de vue logique

- a) Création d'un multigraphe à partir d'une molécule
- b) Création d'un multigraphe à partir d'un spectre de bandes
- c) Fusion, fission, et création de fichiers de données pour les programmes POPINF et PLUTO
- d) Visualisation d'un multigraphe, zoom d'une partie de multigraphe, superposition de plusieurs multigraphes
 - Quelques définitions
 - Méthodes utilisées
- e) Interversion de deux points d'un multigraphe et rotation de multigraphe

2) Point de vue implémentation

- a) 3 programmes
- b) Overlay
- c) Description du programme graphique DESSIN
- d) Description du programme graphique HRDCPY

VII. Manuel d'utilisation.

1) Programme FICPRO

- a) Option 1: création d'un multigraphe à partir d'une molécule

- b) Option 2: création d'un multigraphe à partir d'un spectre
- c) Option 3: fusion de fichiers
- d) Option 4: fission de fichiers
- e) Option 5: fichier de données pour POPINF
- f) Option 6: fichier de données pour PLUTO

2) Programme DESSIN

3) Programme HRDCPY

VIII. Evaluation et perspectives.

- 1) Limite de la généralisation des problèmes graphiques
- 2) Librairie GRALIB
- 3) Programmes de visualisation

Bibliographie et Annexe.

CHAPITRE I

Introduction générale:

Le graphisme et la chimie.

En chimie, tout comme dans d'autres sciences expérimentales, on dispose de nombreuses méthodes d'analyse qui permettent de rassembler des mesures; le premier travail du chimiste sera d'ordonner ces résultats. Deux types de présentations sont fréquemment utilisés, soit un tableau représentant des valeurs numériques, soit un graphique. Inutile de dire que la seconde méthode est plus parlante, à tel point que certains problèmes sont caractérisés par une courbe type de résultats.

Souvent aussi, la mesure est réalisée par un appareil d'analyse qui présente directement l'ensemble des résultats sous la forme d'un graphique, et on parlera alors souvent de "spectre". Il suffit de se promener dans un laboratoire expérimental pour constater que de tels appareils ne sont pas rares. Citons pour exemples les appareils de spectroscopie à IR ou UV, la RMN, la chromatographie, l'ESCA et la liste est loin d'être exhaustive. On constate donc que le graphique en tant que visualisation des résultats d'une analyse est monnaie courante dans la vie du chimiste.

Un autre problème qui se rencontre en chimie est la représentation des molécules. A cet effet, certaines méthodes d'analyse suivies d'un traitement des mesures recueillies permettent d'obtenir pour des molécules des paramètres tels que longueur de liaison entre atomes, angles de liaison, angles de torsion, distances d'atomes à un plan défini par d'autres atomes, etc. Ces renseignements sont très précieux pour la reconstruction de la molécule. Mais dans notre esprit, il est difficile de pouvoir l'imaginer à trois dimensions à partir de la seule donnée de ces paramètres, et il nous faut faire appel à des procédés de visualisation plus concrets.

On pourrait simplement se munir de tiges symbolisant les liaisons chimiques et de boules représentant les atomes, et à la manière d'un jeu de construction, il nous resterait à les attacher ensemble tout en respectant bien les paramètres internes de la molécule (c'est à dire longueurs et angles de liaison, et angles de torsion). Cependant, ce travail est ardu, long, peu pratique et l'assistance de l'ordinateur et du graphisme devient ici très précieuse.

Les programmes graphiques peuvent en effet à partir des coordonnées cartésiennes de chaque atome reconstituer exactement la molécule et la présenter à l'écran ou sur papier. Cette image à deux dimensions peut même se compléter d'une vue stéréoscopique permettant à l'utilisateur de voir la molécule à trois dimensions au moyen de lunettes spéciales. Beaucoup peuvent se demander s'il ne s'agit pas là de considérations esthétiques et inutiles. Pour le chimiste, il n'en est rien!

En effet, de nombreuses recherches chimiques se basent sur la connaissance de la structure moléculaire, notamment toutes les études de relation entre les propriétés géométriques et d'autres facteurs tels que l'activité pharmacologique, l'activité enzymatique, les propriétés électroniques,.... Pour ce genre de travaux, le passage des propriétés géométriques numériques à la représentation de la conformation moléculaire à l'écran (et/ou sur papier) constitue un gros avantage car il permet au chimiste de percevoir directement certaines caractéristiques structurales comme par exemple la forme générale de la molécule (étendue ou repliée), l'importance de la surface de contact, la sortie de groupements fonctionnels hors du plan de la molécule, la situation de certains groupes par rapport à d'autres,.... En outre, des améliorations

sont toujours possibles afin de faciliter encore ces études; par exemple des rotations peuvent présenter la molécule sous différents angles, des superpositions aident parfois à mettre en évidence des différences géométriques entre deux molécules de même famille,...

L'ordinateur et le graphisme apportent donc une aide non négligeable aux chimistes dans leurs travaux de recherche et notre travail consistera à implémenter un outil interactif permettant de résoudre deux problèmes graphiques particuliers.

CHAPITRE II

Introduction aux problèmes graphiques

à l'origine de ce travail.

Le premier problème graphique qui sera traité dans ce mémoire est la représentation de structures moléculaires et, plus particulièrement, la représentation de systèmes polymériques.

Le second problème est plus spécifique aux chimistes théoriciens qui étudient les polymères: il s'agit de l'ordonnement d'une structure de bandes d'énergie électronique d'un polymère.

Qu'est-ce qu'une structure de bandes?

Dès que deux atomes sont mis en présence pour former une liaison chimique, il apparaît directement un dédoublement du niveau fondamental des particules, l'un d'énergie plus basse que le niveau fondamental, l'autre d'énergie plus élevée.

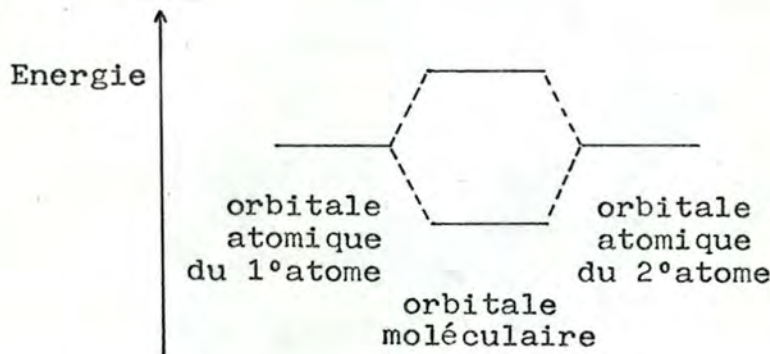


Fig. II.1

Si au lieu de deux atomes en présence, il y en a N (cristal, molécule infinie), nous obtenons alors N niveaux. La distinction entre niveaux discrets n'est plus possible mais il apparaît une distribution quasi continue appelée bande d'énergie.

Chaque polymère est ainsi caractérisé par un spectre de bandes qui consiste en un graphique ayant l'énergie pour ordonnée et une abscisse particulière appelée point k .

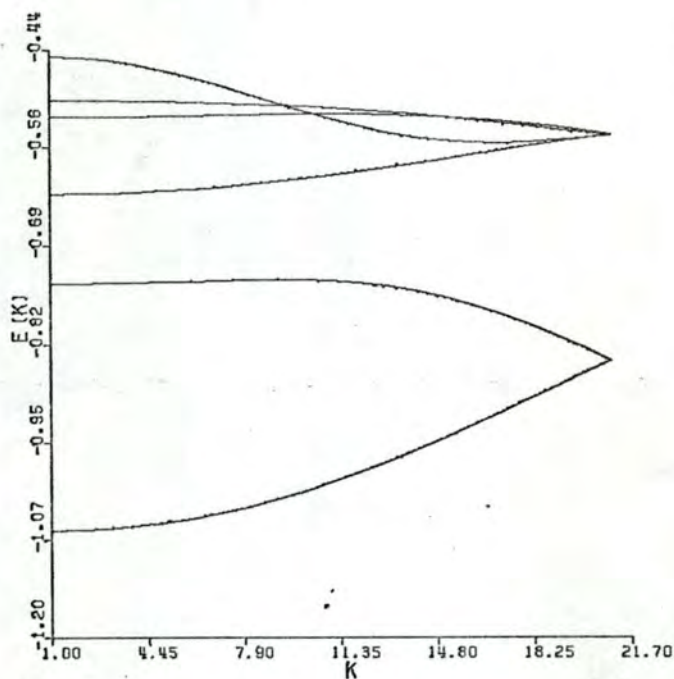


Fig. II.2 : Structure de bandes du polyéthylène.

Qu'est-ce que l'ordonnement d'une structure de bandes?

En fait, les valeurs énergétiques sont fournies par un programme de calcul. Une des étapes de ce programme est d'attribuer les valeurs énergétiques aux différentes bandes. Pour cela, il attribue les n premières valeurs de plus faible énergie à la première bande, puis les n suivantes à la seconde bande, et ainsi de suite jusqu'au moment où tous les points ont été attribués; n représente le nombre de points en abscisse pour lesquels les calculs ont été réalisés.

La conséquence évidente de cette procédure est que les bandes ne se croiseront pas, ce qui ne représente malheureusement pas la réalité chimique. Dès lors, il faut avoir la possibilité de corriger cette structure de bandes originale avant de l'injecter dans la suite des programmes de calcul. Cette étape de modification constitue l'ordonnancement du spectre de bandes.

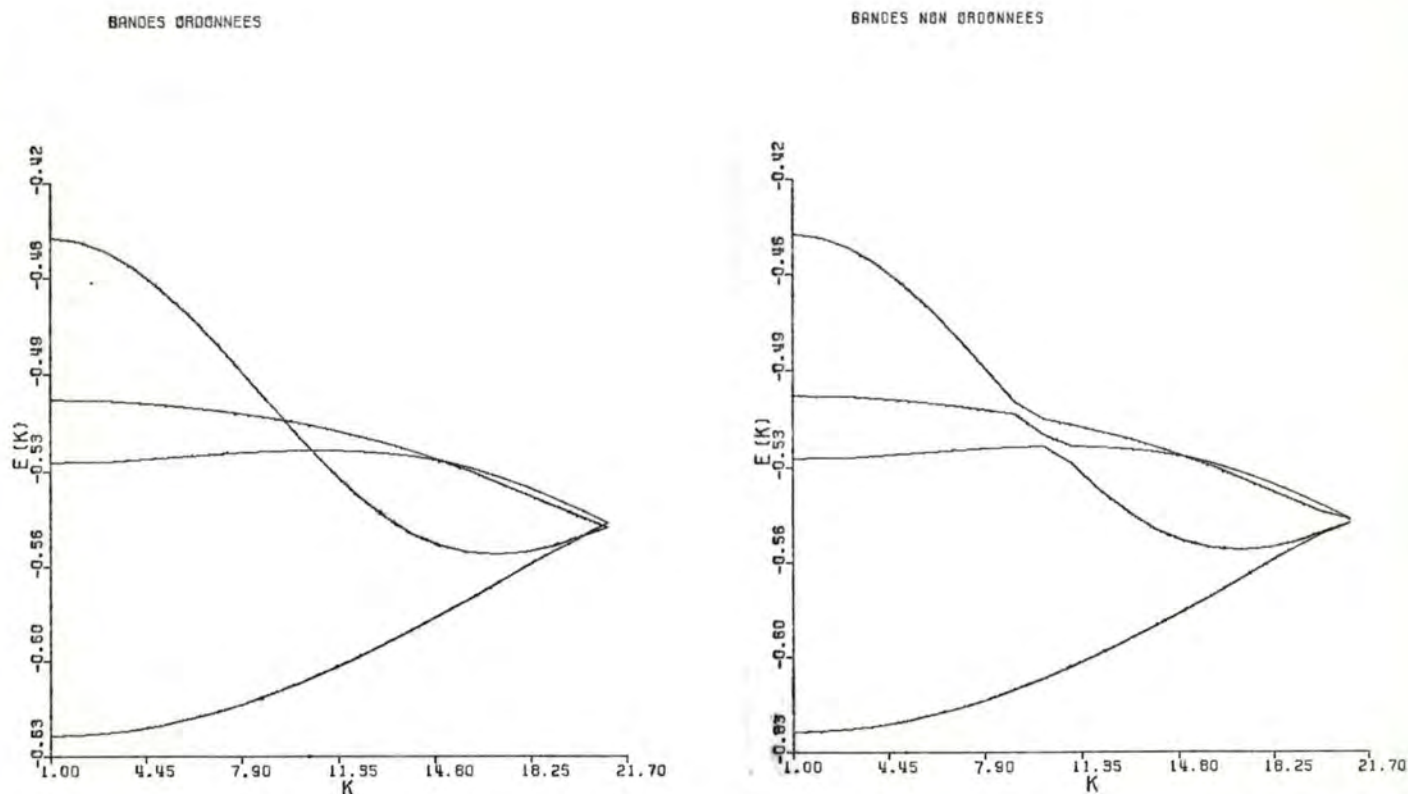


Fig. II.3

CHAPITRE III .

Etude de la configuration et de l'architecture
du système graphique.

Avant de discuter des problèmes du mémoire, nous allons décrire brièvement l'environnement dans lequel le travail a été réalisé.

1) Configuration.

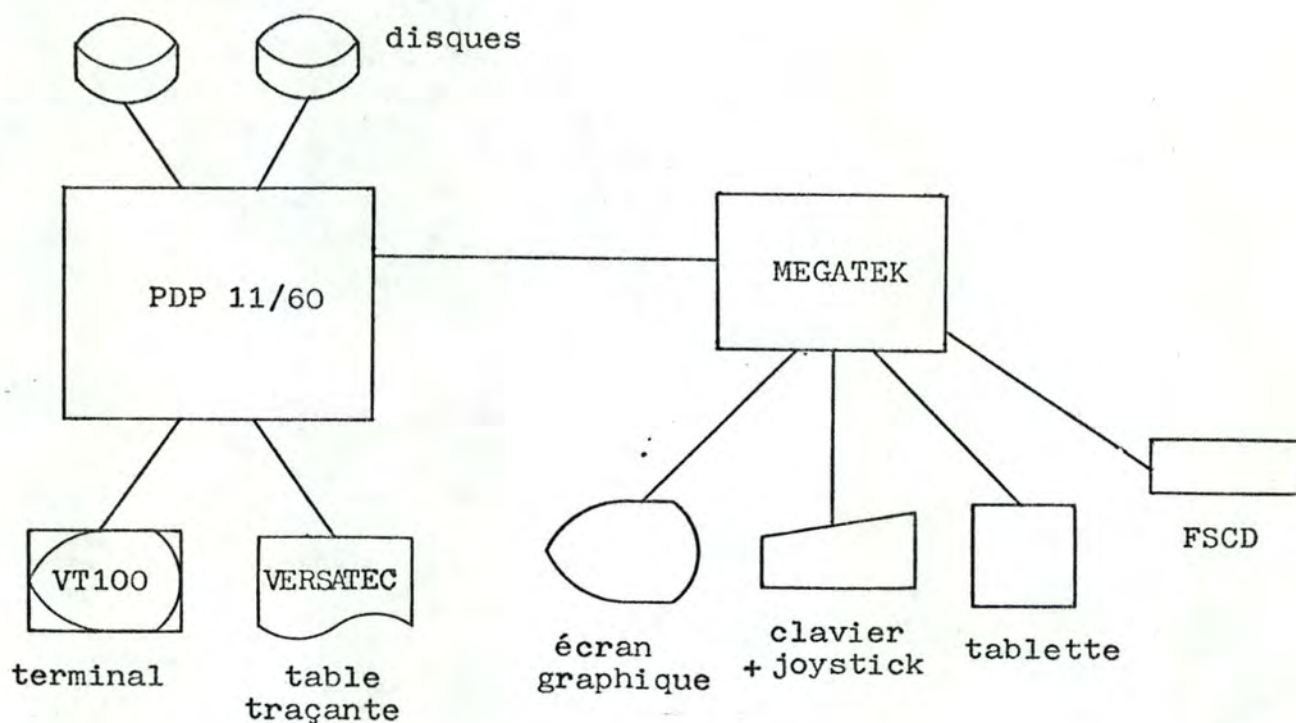


Fig. III.1

L'équipement comprend:

- un ordinateur PDP 11/60 possédant une mémoire de 64 K mots de 16 bits et deux disques de 5 Mbytes,
- un terminal conversationnel VT100,
- une table traçante Versatec,

• un système graphique comprenant

un écran graphique d'une résolution de 4096*4096

un clavier avec joystick

une tablette avec souris

une boîte de fonction (FSCD = Function Switch and Control Dials) avec 8 potentiomètres et 16 boutons programmables.

Le software de base se compose:

- du système d'exploitation RSX-11M

- des logiciels graphiques { WAND 7200 [1]
VERSAPLOT [2]

2) Architecture du Megatek. [3]

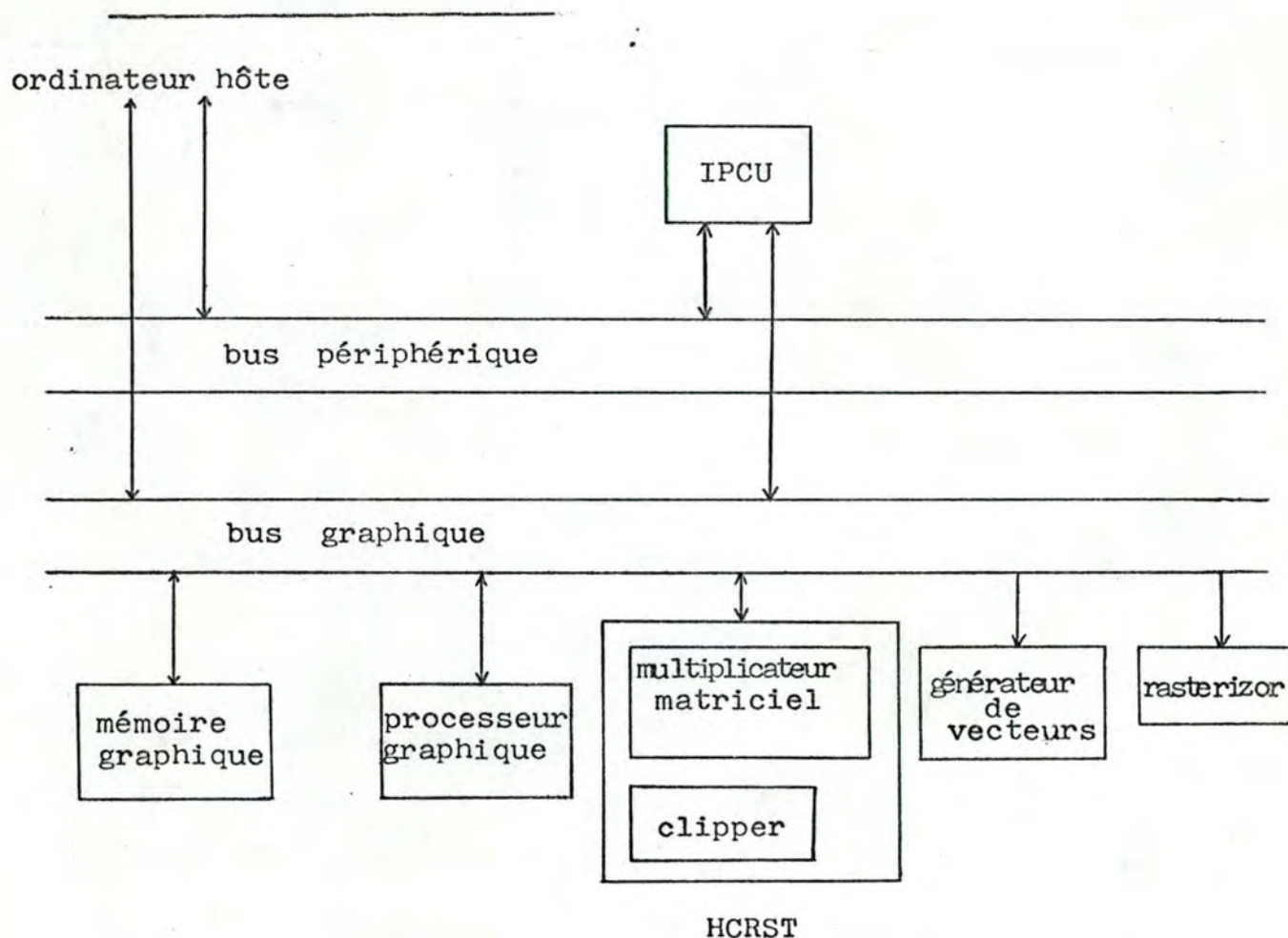


Fig. III.2

Bus

Ils sont au nombre de deux et permettent la communication entre les différents modules.

Le bus graphique s'occupe de véhiculer les commandes de contrôle de l'image, de la mémoire graphique vers les différents modules graphiques (c'est à dire processeur, HCRST, générateur, rasterizer).

Le bus périphérique s'occupe de la communication entre les différents périphériques de l'ordinateur.

Une telle organisation permet de s'occuper localement des périphériques du système Megatek sans interférer avec le traitement de la régénération des dessins.

Mémoire graphique

La capacité de la mémoire est de 16 K mots de 32 bits. Les 256 premiers mots sont réservés et le reste est disponible pour les listes d'affichage, c'est à dire des ensembles d'instructions de contrôle de l'image qui constituent de véritables programmes exécutables.

Processeur graphique

Il s'agit d'un microprocesseur qui contrôle les accès à la liste d'affichage stockée dans la mémoire.

Ce processeur lit séquentiellement dans la mémoire. Pour chaque mot, il va examiner les bits de fonction et décider de la

manière d'interpréter le reste du mot; finalement, il fournira l'information correspondante au générateur de vecteurs.

Quand le processeur atteint la fin de la liste d'affichage, il retourne au début de celle-ci et recommence la lecture de la liste. Ce traitement est réalisé 50 fois par seconde.

Générateur de vecteurs

Les dessins affichés sont composés entièrement de vecteurs (traits). Ce générateur a pour rôle de convertir l'information au sujet des coordonnées (X,Y) en un signal lumineux créé par déplacement d'un faisceau sur l'écran.

Le faisceau tracera un trait quand il est en position allumé (DRAW), et déplacera sa position courante quand il est en position éteinte (MOVE). Un vecteur peut être tracé suivant deux modes:

un vecteur absolu spécifie les coordonnées (X,Y) finales de la ligne à tracer. Le point de départ est la position courante du faisceau;

un vecteur relatif spécifie le déplacement suivant les deux axes ($\Delta X, \Delta Y$) que le faisceau doit parcourir;

dans les deux modes, les coordonnées finales représentent la nouvelle position courante.

HCRST = Hardware Clip, Rotate, Scale and Translate

Les opérations de rotation, de mise à l'échelle et de translation nécessitent de recalculer les coordonnées. Si ces opérations sont réalisées par software, cela aura pour effet d'alourdir le travail de l'ordinateur. Le module HCRST permet de réaliser ces opérations par hardware.

Pratiquement, le programme construit une matrice de transformation et à l'aide d'un appel à une sous-routine, il la transmet à la mémoire du système graphique. Le processeur graphique fournira cette matrice au module HCRST.

Ce module est constitué de deux parties. La première applique la matrice de transformation aux différents vecteurs concernés. La seconde partie (Clipper) a pour effet de supprimer les vecteurs qui seraient hors de l'écran.

Rasterizer = Dot Matrix Plotter Interface

Ce module permet à l'utilisateur de générer sur papier, à l'aide d'une table traçante, une copie du dessin visualisé à l'écran. Il s'agit d'un module hardware dont le rôle est de convertir les vecteurs représentés à l'écran en une série de points qui seront tracés par le marqueur de la table traçante.

Il faut faire remarquer que ce module est absent dans l'architecture du système graphique existant à l'Institut.

IPCU = Intelligent Peripheral Control Unit

Le microprocesseur situé dans l'IPCU est connecté aux bus graphique et périphérique afin de réaliser une interface intelligente entre le système graphique et les différents périphériques, c'est à dire le clavier, la tablette et le joystick.

3) Exécution d'un programme graphique.

Dans ce paragraphe, nous allons donner une explication sommaire des interactions entre le PDP 11/60 et le Megatek lors de l'exécution d'un programme graphique.

Les programmes d'application sont stockés dans la mémoire du PDP 11/60 et exécutés par celui-ci. Chaque fois qu'un appel à une routine graphique est rencontré dans le programme, le processeur du PDP écrit les instructions correspondantes dans la mémoire graphique. La suite d'instructions dans cette mémoire graphique constitue une liste d'affichage qui représente un véritable programme exécutable. La liste d'affichage sera parcourue par le processeur graphique qui réalisera les différentes opérations (tracé de trait, déplacement de la position courante, clignotement d'un dessin, suppression d'un dessin). La liste d'affichage sera complétée et modifiée au fur et à mesure de l'exécution du programme d'application.

CHAPITRE IV .

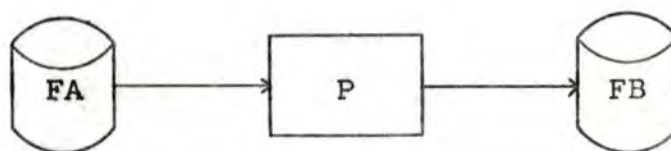
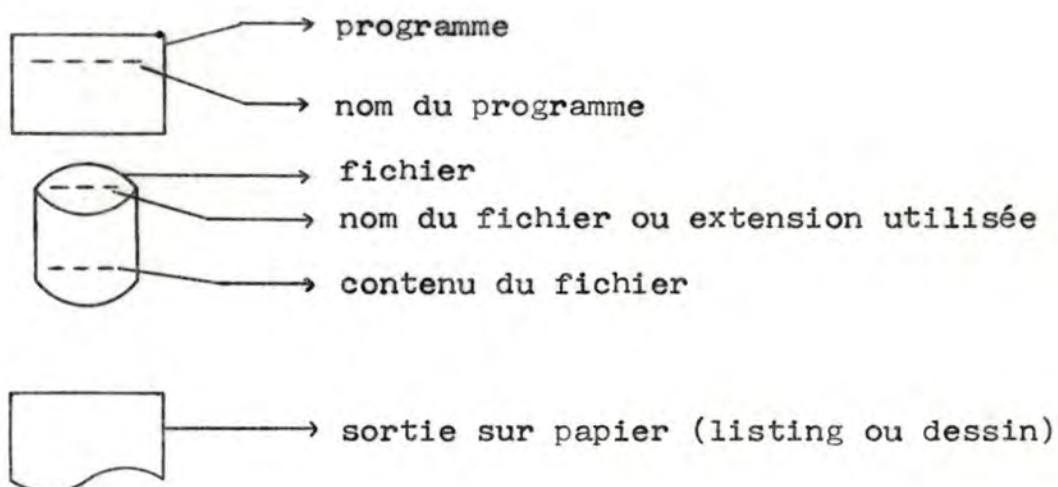
Analyse d'opportunité.

1) Analyse de l'existant.

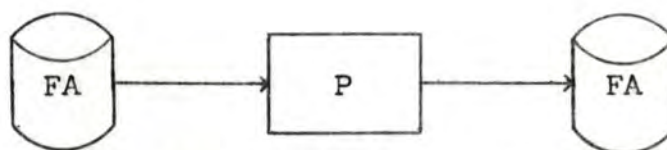
Le laboratoire de Chimie Théorique Appliquée dispose d'une chaîne de programmes qui permet d'étudier la structure géométrique et électronique des polymères. Cette chaîne comporte différents programmes de calcul (Geomol, Ehco81, Popinf, Madene) et d'autres à tendance graphique (Pluto, Ordbae, Dess94).

Description de la chaîne de programmes.

a) Conventions utilisées dans les schémas.



le programme P utilise les données du fichier FA et crée le fichier FB sans modifier FA



le programme P utilise les données du fichier FA et modifie celui-ci.

Fig. IV.1

b) Schéma de la chaîne de programmes.

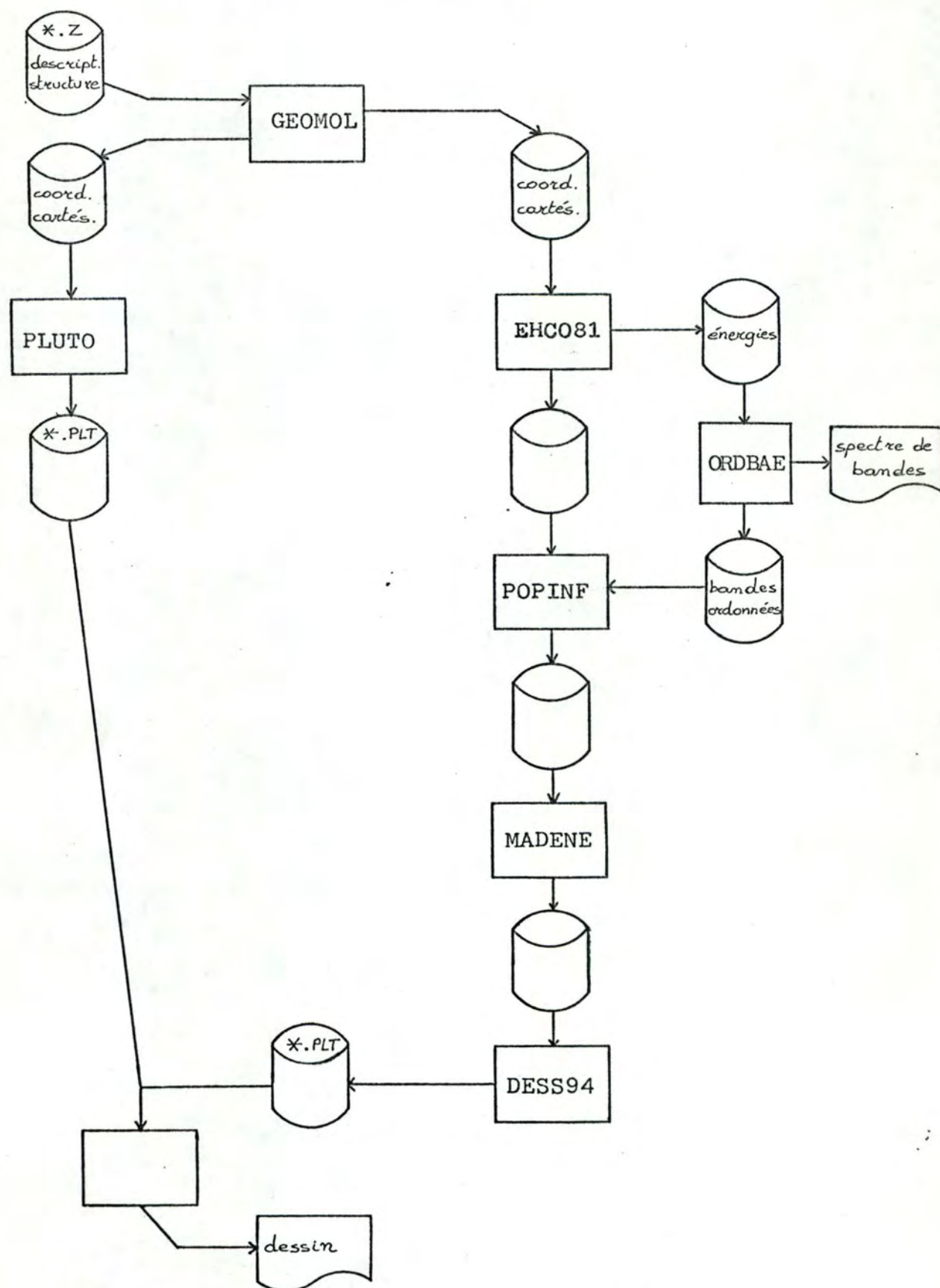


Fig. IV.2

Le schéma proposé ci-dessus reprend l'ensemble des programmes utilisés dans la chaîne. Cependant, comme ce mémoire est un travail concernant le graphisme, seuls les programmes à tendance graphique seront détaillés. Nous négligerons dès lors l'explication de la théorie mathématique, support des programmes de calcul, et nous renvoyons le lecteur à la référence [4] pour de plus amples informations.

c) Description des programmes graphiques.

1° programme: PLUTO [5]
.....

Rôle: représentation d'une molécule sous différents angles d'observation et suivants différents tracés.

Entrée: fichier avec coordonnées cartésiennes des atomes de la molécule.

Sortie: fichier binaire contenant les informations pour représenter le dessin.

Utilisation du programme:

après avoir spécifié le nom du fichier de données, l'utilisateur doit entrer différents ordres qui lui permettent de définir les caractéristiques du dessin, c'est à dire

type de tracé: avec tiges

avec tiges et boules

avec sphères de Van der Waals

type de représentation: stéréo ou non

avec ou sans ombrage

angle d'observation
dimensions du dessin,...

L'exécution du programme a pour effet de créer directement un fichier; celui-ci servira d'entrée à un programme utilitaire qui produira le dessin via une table traçante.

2° programme: ORDBAE
.....

Rôle: permettre à l'utilisateur, via un dialogue, d'ordonner les bandes d'un spectre.

Entrée: fichier avec les énergies.

Sortie: fichier avec les numéros des bandes ordonnées; représentation du spectre de bandes sur papier listing.

Utilisation du programme:

ce programme nécessite l'emploi d'une télétype. Après avoir défini les caractéristiques du spectre de bandes (nombre de points k, nombre de bandes), l'utilisateur peut réaliser l'ordonnancement en introduisant pour chaque couple de points à intervertir, le numéro du point k et les deux numéros de bandes concernées.

3° programme: DESS94
.....

Rôle: représentation du spectre de bandes et de deux autres graphiques caractéristiques du polymère.

Entrée: fichier créé par les programmes de calcul de la chaîne.

Sortie: fichier binaire contenant les informations pour représenter le dessin.

Utilisation du programme:

l'exécution du programme a pour effet de créer directement un fichier; celui-ci servira d'entrée à un programme utilitaire qui produira le dessin via une table traçante.

2) Critique de l'existant.

En ce qui concerne la représentation des molécules, les causes d'insatisfaction sont les suivantes:

le programme Pluto n'utilise pas d'écran graphique comme périphérique de sortie, dès lors le dessin sur papier est le seul moyen de visualisation. Si par malheur l'orientation de la molécule n'est pas satisfaisante, l'utilisateur est obligé de produire plusieurs dessins avant d'obtenir ce qu'il désire;

l'utilisation des instructions permettant de réaliser les rotations est peu aisée à cause de leur nombre et des différents référentiels employés; en effet, pour effectuer la rotation, il faut introduire les valeurs des paramètres angulaires à l'aide du clavier.

Pour ce qui est de l'ordonnancement des bandes, il y a lieu de souligner les points suivants:

- la difficulté de visualisation, car le programme ne représente pas le spectre par un tracé continu mais seulement avec les numéros de bandes,
- la difficulté de réaliser l'ordonnancement vu l'obligation d'utiliser une télétype,
- la consommation importante de papier listing pour permettre la représentation du spectre,
- la difficulté de contrôler les modifications à cause du type de représentation.

3) Expression des besoins.

Tout d'abord, il faut préciser que la situation a changé récemment sur le plan de l'équipement; en effet, le Département de Chimie a maintenant accès à un matériel graphique de haute performance dont l'utilisation devrait permettre de résoudre les problèmes relevés au point précédent.

a) Objectifs généraux.

- La représentation des structures moléculaires à l'écran graphique avec possibilité d'appliquer de manière interactive des opérations telles que rotation, translation, mise à l'échelle,...

-La conception et l'implémentation d'un outil interactif permettant de réaliser l'ordonnancement des bandes de manière aisée.

-La possibilité d'obtenir un dessin, sur table traçante, de l'objet graphique visualisé à l'écran.

b) Ebauche de solutions.

-Pour la représentation des molécules:

Dans la chaîne existante, le logiciel de base pour la représentation des molécules est le programme Pluto. Ce programme a l'avantage d'offrir beaucoup d'options de travail, mais il est impossible de l'implémenter tel quel sur le PDP 11/60 à cause de la taille limitée du PDP et de l'absence de structure de recouvrement dans ce programme.

Dès lors, nous avons été amenés à scinder en deux parties le problème de la représentation des molécules.

Dans une première partie, où l'on travaillera avec le système Megatek et le PDP, les options peu gourmandes en place mémoire seront implémentées, c'est à dire

- la représentation de la molécule avec des tiges, avec des tiges et les symboles atomiques
- les rotations de la molécule.

Dans la seconde partie, on utilisera le programme Pluto (implémenté sur le Dec) afin de disposer des options de représentation plus sophistiquées telles que

- la représentation de la molécule avec des tiges et des boules, des sphères de Van der Waals,
- la représentation de la molécule en stéréoscopie, avec ombrage.

-Pour l'ordonnancement des bandes:

L'outil à réaliser devra répondre aux exigences concrètes suivantes:

- la représentation de l'ensemble du spectre avec un système d'axes
- la possibilité de sélectionner une zone du spectre à l'aide d'une fenêtre, et de l'agrandir
- la représentation à l'aide d'un tracé continu ou à l'aide des numéros de bandes
- la possibilité d'indiquer facilement et de manière interactive les points du spectre à intervertir
- la possibilité d'accéder aux coordonnées d'un point quelconque du spectre.

-Pour la représentation à la table traçante:

L'étude de l'architecture du système graphique a permis de constater l'absence d'un module hardware (Rasterizer), celui-ci ayant pour rôle de produire sur papier le dessin visualisé à l'écran.

Or, la possibilité d'obtenir un dessin sur papier de l'objet graphique présenté à l'écran, entre dans les spécifications de ce travail.

Dans un premier temps, la solution suivante a été proposée pour pallier à cette absence d'hardware.

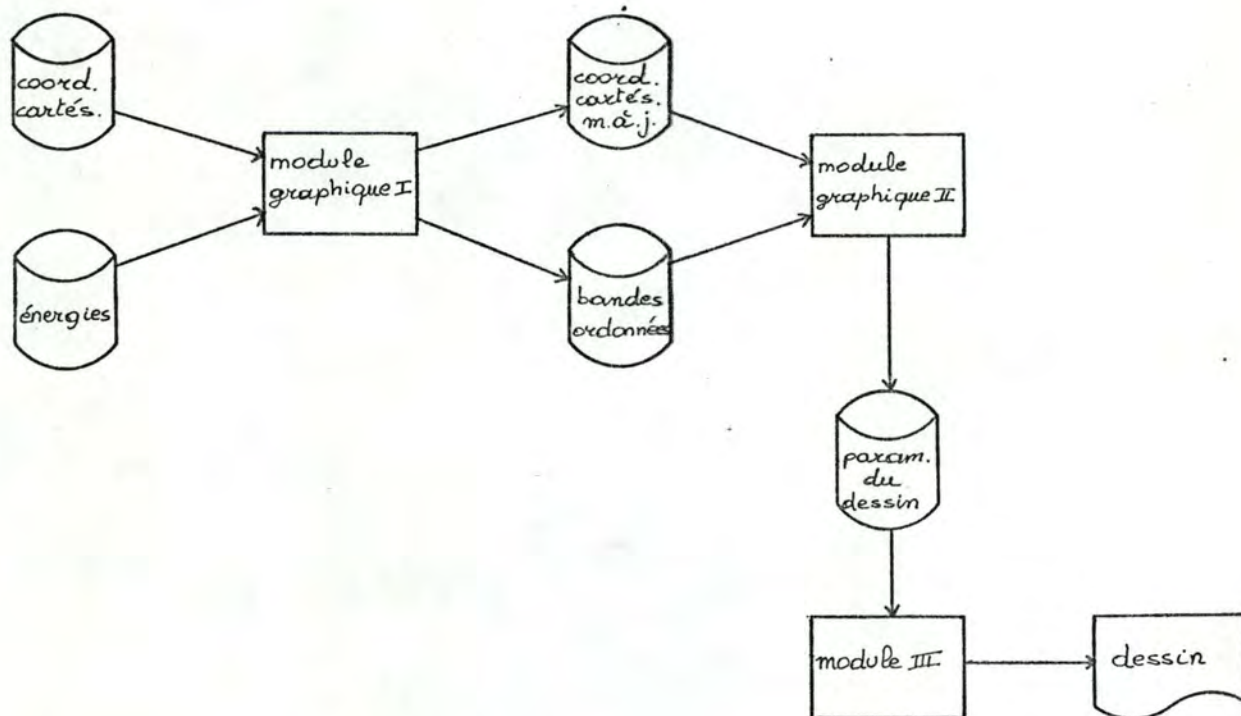


Fig. IV.3

Le module graphique I travaillerait avec le Megatek, tandis que le module graphique II aurait pour but de générer le dessin en utilisant un logiciel graphique (DI3000, Versaplot,...). Le module III est le programme utilitaire qui, à partir d'un fichier binaire, produit le dessin à la table traçante.

Cependant, il faut être bien conscient qu'avec une telle liaison système graphique-table traçante, nous n'aurons pas de solution standardisée, mais une solution valable uniquement dans le cadre de nos applications (c'est à dire représentation de molécules et de spectres de bandes).

Or, il est certainement intéressant d'essayer de sortir du cadre de ces deux applications et d'aboutir à une certaine généralisation des problèmes graphiques.

CHAPITRE V

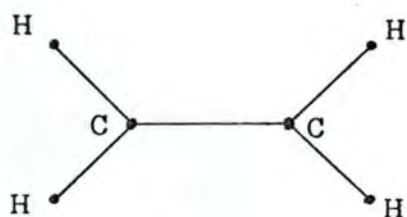
Proposition de standardisation des problèmes graphiques.

En étudiant les deux applications envisagées dans le mémoire (structures moléculaires et ordonnancement des bandes), on constate qu'à la sortie des modules de calcul respectifs (Geomol et Ehco81), on dispose dans un cas des coordonnées cartésiennes des atomes de la structure, et dans l'autre des valeurs d'énergie pour différents points k du spectre de bandes.

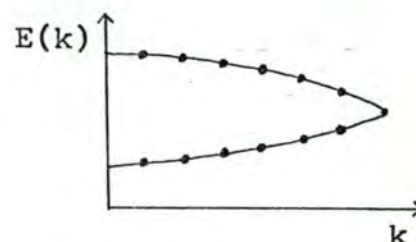
De plus, pour être à même de représenter une molécule, il faut avoir défini l'ensemble des liaisons entre ses atomes.

1) Principe.

On peut donc considérer que dans les deux cas, nous avons des points dont certains sont reliés entre eux.



Molécule d'éthylène



Spectre de bandes

Fig. V.1

D'où l'idée de considérer n'importe quel objet graphique comme étant constitué d'un ensemble de sommets et d'un ensemble d'arêtes (c'est à dire un multigraphe).

On a ainsi une représentation standardisée des objets graphiques, et les différents modules graphiques travailleront exclusivement sur des multigraphes.

2) Moyen.

Pour réaliser le principe énoncé ci-dessus, il faut mettre au point des primitives qui permettent de créer et de modifier un multigraphe. Les opérations indispensables sont les suivantes:

- créer un sommet,
- accéder aux informations d'un sommet,
- modifier les informations d'un sommet,
- créer une arête,
- accéder aux informations d'une arête,
- modifier les informations d'une arête.

De plus, il faut choisir une structure de données pour mémoriser les informations concernant un multigraphe. Nous allons d'abord définir les informations nécessaires qui décrivent un multigraphe avant de discuter du choix de la structure de données.

a) Informations nécessaires pour la description d'un élément du multigraphe (sommet ou arête).

Cet ensemble d'informations nécessaires a été défini en fonction des deux applications étudiées dans ce travail. Pour décrire un sommet, on dispose

- 1°) d'une zone 'label'
- 2°) d'une zone 'numéro du sommet'
- 3°) d'une zone 'coordonnée X'
- 4°) d'une zone 'coordonnée Y'
- 5°) d'une zone 'coordonnée Z'
- 6°) d'une zone 'caractéristique'

Pour décrire une arête, on dispose

- 1°) d'une zone 'numéro de l'arête'
- 2°) d'une zone 'numéro du sommet initial'
- 3°) d'une zone 'numéro du sommet final'
- 4°) d'une zone 'caractéristique'

Il faut savoir qu'aucune des deux applications n'utilisent l'ensemble des zones décrites. Il reste donc de la place disponible pour enregistrer des données complémentaires lors de développements ultérieurs de ces applications.

Il nous semble important de faire ici l'avertissement suivant. Cet ensemble de zones permet déjà de décrire les objets graphiques rencontrés dans de nombreuses applications. Il se peut cependant que pour une application particulière, il soit indispensable d'enregistrer plus de données qu'il n'existe de zones réservées. La solution sera donc d'ajouter une (ou plusieurs) zone(s) immédiatement à la suite de celles déjà existantes afin de ne pas modifier la logique de fonctionnement du système de mémorisation.

b) Numérotation utilisateur et numérotation système.

Chaque utilisateur souhaite habituellement posséder ses propres indices pour identifier les éléments d'un graphe; d'une façon générale, il utilisera un chiffre. Mais la séquence des numéros sera souvent aléatoire, ce qui ne peut servir efficacement à la mémorisation des sommets et des arêtes. Dès lors, afin de ne pas imposer à l'utilisateur une numérotation des éléments du graphe, le système de mémorisation aura la sienne et il devra réaliser la correspondance entre numérotation utilisateur et numérotation système.

c) Choix d'une structure de données.

Dans un premier temps, les deux solutions suivantes ont été envisagées: -utilisation de tableaux

-utilisation d'un fichier à
accès direct.

Etudions les inconvénients et les avantages de chaque solution.

-Utilisation de tableaux:

Avec cette solution, le système de mémorisation fonctionnerait comme suit. La correspondance entre numéro utilisateur et numéro système serait réalisée à l'aide d'une table. L'indice du tableau représenterait le numéro système. La création d'un sommet (ou d'une arête) consisterait en des affectations de valeurs à des cases de tableaux.

Les avantages de cette solution sont l'accès immédiat aux informations et le fait qu'on peut créer une arête dès que ses sommets ont été définis.

Par contre, on dénombre deux inconvénients majeurs. D'abord, on est obligé de dimensionner les tableaux en fonction des applications importantes (qui ne sont pas nécessairement les plus fréquentes), d'où une perte de place en mémoire centrale. Essayons d'estimer cette perte de place. L'application la plus importante à envisager est l'ordonnancement des bandes. Ce problème peut traiter jusqu'à 150 bandes de 21 points, soient 3150 points (ou sommets) et pratiquement autant d'arêtes. Or actuellement, pour un sommet, il existe six zones mémoire réservées, donc pour 3150 points il faudrait un tableau de dimension (3150 X 6). Pour une arête, il existe quatre zones réservées, donc pour 3150 arêtes il faudrait un tableau de dimension (3150 X 4). Au total, 31500 zones mémoire sont donc réservées.

D'un autre côté, la plupart des applications (molécules, fonctions, polymères) auront un nombre de sommets souvent inférieur à 100. La place utilisée sera donc de 1000 cases, ce qui représente un taux d'occupation de 3.2%, soient 30500 zones mémoire non utilisées.

Le second inconvénient est que l'utilisation de tableaux a une conséquence à plus long terme sur le regroupement des modules. En effet, elle oblige de regrouper en un seul programme tous les modules qui travaillent avec le système de mémorisation. Si tel n'était pas le cas, il faudrait sauver l'ensemble des informations relatives à un multigraphe sur un fichier temporaire, et passer au programme suivant dont la première tâche serait de recharger les informations dans les tableaux.

-Utilisation d'un fichier à accès direct:

Avec cette solution, la correspondance entre numéro utilisateur et numéro système serait aussi réalisée à l'aide d'une table. Le numéro de l'enregistrement représenterait le numéro système. La création d'un sommet (ou d'une arête) consisterait en une écriture d'enregistrement en fin de fichier.

L'avantage principal est une perte de place minimale. En effet, la taille de l'enregistrement doit être définie en fonction de l'information la plus longue, c'est à dire la description d'un sommet qui nécessite six zones mémoire.

L'inconvénient de cette solution est qu'il faut d'abord définir tous les sommets, puis les arêtes, afin de garder un système de numérotation simple. Un autre inconvénient est que le nombre d'entrée/sortie risque de devenir

un facteur contraignant avec une application importante.

Nous venons d'étudier deux solutions, toutes deux présentant des inconvénients importants qui n'incitent pas à les retenir. Nous allons voir s'il est possible de les améliorer afin de dégager une solution qui réaliserait le système de mémorisation avec suffisamment d'efficacité.

Comment pallier au problème du surdimensionnement des tableaux? On pourrait se limiter aux 20 premières bandes du spectre (c'est à dire les plus intéressantes), ou morceler l'ensemble du spectre par paquets de 10 bandes; ceci offre l'avantage de conserver la description complète du spectre. Ces deux solutions permettraient de revenir à des dimensions plus raisonnables pour les tableaux lors de l'enregistrement du multigraphe. Mais pour représenter à l'écran le système complet, le module graphique doit disposer de l'ensemble des informations. Dès lors, le problème du surdimensionnement est simplement déplacé à l'étape suivante du travail.

Avec l'utilisation d'un fichier à accès direct, la contrainte la plus importante est d'obliger l'utilisateur à définir d'abord tous les sommets, puis seulement les arêtes. Pour supprimer cet inconvénient, il suffit d'utiliser deux fichiers à accès direct, l'un contenant la description des sommets et l'autre la description des arêtes. Cette solution permet aussi de définir la taille optimale pour l'enregistrement et il n'y a donc plus de perte de place.

De plus, afin que le nombre d'entrée/sortie ne soit pas un facteur contraignant, on donnera la possibilité à l'utilisateur de morceler un spectre de bandes impor-

tant en plusieurs multigraphes de tailles plus raisonnables; cependant, pour visualiser le spectre complet, il faudra implémenter une procédure de superposition de multigraphes.

Il faut savoir qu'avec cette solution utilisant deux fichiers à accès direct, il existe encore un problème de surdimensionnement. Mais celui-ci est limité aux tableaux réalisant la correspondance entre numérotation utilisateur et numérotation système. Le taux d'occupation est le même (3.2%) mais le nombre de places non utilisées est beaucoup moindre (3050 sur 3150).

Finalement, cette dernière solution a été retenue parce qu'on a su supprimer les inconvénients principaux et aussi parce qu'elle n'impose aucune contrainte au niveau du regroupement des modules qui seront développés ultérieurement.

3) Librairie GRALIB.

Pour pouvoir utiliser le système de mémorisation, il faut mettre au point une librairie de routines dont l'utilisateur ne connaîtra que les effets. Ces routines devront permettre de créer les éléments d'un multigraphe, d'accéder aux informations concernant un multigraphe existant, et de modifier ces informations.

Un jeu minimal de routines a donc été développé en fonction des deux applications particulières de ce mémoire. Notre but n'a pas été de fournir un ensemble complet de routines, il est donc certain que d'autres applications nécessiteront la création de nouvelles routines (par exemple la suppression d'un élément du graphe).

a) Fonctionnement_général_de_la_librairie.

Le système de mémorisation utilise deux fichiers à accès direct, l'un appelé GSOM.DAT utilisé pour la description des sommets, l'autre appelé GART.DAT qui contient la description des arêtes.

La correspondance entre numérotation utilisateur et numérotation système est réalisée à l'aide de deux tableaux unidimensionnels (un pour les sommets, l'autre pour les arêtes). On mémorise dans la case du tableau le numéro utilisateur et la valeur de l'indice de cette case représente le numéro de l'enregistrement où la description complète du sommet ou de l'arête est stockée.

L'enregistrement des éléments du graphe commence au record n°2. Le premier record sert à stocker le nombre total de sommets et un titre dans le cas du fichier des sommets, et le nombre total d'arêtes dans le cas du fichier des arêtes.

b) Règles_d'utilisation.

- Avant de commencer à définir un graphe, il faut absolument "ouvrir" le système de mémorisation [DEBUT] .
- Quand le graphe est défini, il faut absolument clôturer le système de mémorisation [FIN] .
- Lorsqu'on désire utiliser un graphe déjà défini (et clôturé) ou compléter un graphe partiellement défini, il faut, après ouverture, remplacer la table de correspondance dans son état lors de sa dernière clôture [INIGRA] .
- Avant de définir une arête, il faut définir ses sommets.

-La plupart des routines renvoient un code erreur qui renseigne sur la manière dont la routine s'est exécutée.

c) Description des différentes routines existantes.

Subroutine DEBUT (label)

rôle: ouverture du système de mémorisation,
c'est à dire des deux fichiers

paramètre: label = titre du multigraphe; il
est obligatoire et doit contenir
moins de 50 caractères

Subroutine FIN

rôle: clôture du système de mémorisation,
c'est à dire •écriture du premier enre-
gistrement de chaque fichier
•fermeture des fichiers

Subroutine INIGRA

rôle: quand on désire modifier ou compléter un
graphe déjà défini lors d'une session an-
térieure; comprenant une ouverture et une
clôture, cette routine sert à replacer
les tables de correspondance dans leur
état lors de la dernière clôture

Subroutine CRESOM (lab, numsom, coord, detsom, icode)

rôle: créer un sommet

paramètres: lab = label du sommet (2 caractères max)
 numsom = numéro utilisateur du sommet
 (≤9999)
 coord = tableau à 3 cases qui contient
 les coordonnées du sommet
 detsom = caractéristique du sommet
 icode = code erreur (renvoyé par la
 routine)

code erreur = 0 → pas d'erreur
 -1 → il n'y a plus de place pour
 créer un nouveau sommet

Subroutine NUSYSO (numsom, numsys, icode)

rôle: accéder au numéro système d'un sommet à
 partir de son numéro utilisateur

paramètres: numsom = numéro utilisateur dont on
 cherche le numéro système
 numsys = numéro système du sommet
 (renvoyé par la routine)
 icode = code erreur (renvoyé par la
 routine)

code erreur = 0 → pas d'erreur
 -1 → aucun sommet n'a un tel nu-
 méro utilisateur

Subroutine DATASO (numsys, lab, coord, detsom, icode)

rôle: accéder aux données d'un sommet à partir
 de son numéro système

paramètres: numsys = numéro système du sommet
 lab = label (renvoyé par la routine)
 coord = tableau à 3 cases qui contient les coordonnées du sommet (renvoyé par la routine)
 detsom = caractéristique (renvoyée par la routine)
 icode = code erreur (renvoyé par la routine)

code erreur = 0 → pas d'erreur
 -1 → le numéro système ne correspond pas à un sommet

Subroutine CHGES0 (numsys, lab, coord, detsom, icode)

rôle: modifier les données d'un sommet à partir de son numéro système

paramètres: numsys = numéro système du sommet
 lab = nouveau label
 coord = tableau à 3 cases contenant les nouvelles coordonnées du sommet
 detsom = nouvelle caractéristique
 icode = code erreur (renvoyé par la routine)

code erreur = 0 → pas d'erreur
 -1 → le numéro système ne correspond pas à un sommet

Subroutine CREART (numart, numini, numfin, detar, icode)

rôle: créer une arête

paramètres: numart = numéro utilisateur de l'arête
 numini = numéro système du sommet initial
 numfin = numéro système du sommet final
 detar = caractéristique de l'arête
 icode = code erreur (renvoyé par la routine)

code erreur = 0 → pas d'erreur
 -1 → il n'y a plus de place pour
 créer une nouvelle arête

Subroutine NUSYAR (numart, numsys, icode)

rôle: accéder au numéro système d'une arête à
 partir du numéro utilisateur

paramètres: numart = numéro utilisateur de l'arête
 numsys = numéro système (renvoyé par la routine)
 icode = code erreur (renvoyé par la routine)

code erreur = 0 → pas d'erreur
 -1 → aucune arête n'a un tel numéro utilisateur

Subroutine DATART (numsys, numini, numfin, detar, icode)

rôle: accéder aux données d'une arête à partir
 de son numéro système

paramètres: numsys = numéro système de l'arête
 numini = numéro système du sommet
 initial (renvoyé par la
 routine)
 numfin = numéro système du sommet
 final (renvoyé par la rou-
 tine)
 detar = caractéristique (renvoyée
 par la routine)
 icode = code erreur

code erreur = 0 → pas d'erreur
 -1 → le numéro système ne corres-
 pond pas à une arête

Subroutine CHGEAR (numsys, numini, numfin, detar, icode)

rôle: modifier les données d'une arête

paramètres: numsys = numéro système de l'arête
 numini = numéro système du nouveau
 sommet initial
 numfin = numéro système du nouveau
 sommet final
 detar = nouvelle caractéristique
 icode = code erreur (renvoyé par
 la routine)

code erreur = 0 → pas d'erreur
 -1 → aucune arête n'a un tel nu-
 méro système

4) Contraintes inhérentes à la solution implémentée.

Avec l'ensemble des routines décrit ci-dessus, on peut transformer des objets graphiques en multigraphes. Jusqu'à présent, les deux fichiers de travail utilisés contiennent la description d'un et un seul multigraphe, on ne pourra donc visualiser qu'un seul objet graphique à l'écran.

Or, il serait très intéressant de représenter plusieurs polymères à l'écran afin de les comparer, de superposer plusieurs parties d'un spectre de bandes,...

Pour cela, il faut que les fichiers de travail contiennent la description de plusieurs objets graphiques. La solution proposée est d'implémenter un module de fusion qui, à partir de la description de deux objets graphiques (contenue dans 4 fichiers), crée les deux fichiers de travail contenant l'ensemble des informations relatives aux deux objets graphiques. Cette procédure de fusion répétée plusieurs fois permettrait de stocker dans les deux fichiers la description de plusieurs multigraphes.

D'autre part, comme les informations relatives aux multigraphes seront sujettes à des mises à jour lors de l'exécution des modules graphiques (rotation pour molécules, interversion de points du spectre de bandes), il serait intéressant de pouvoir extraire des fichiers de travail une copie de la description d'un multigraphe mis à jour. A cet effet, il faudra disposer d'une procédure de fission.

CHAPITRE VI

Description générale des solutions.

Dans les chapitres précédents, nous avons relevé différents problèmes généraux qu'il faudra résoudre afin de réaliser les objectifs de ce travail. Ces problèmes sont les suivants:

- création d'un multigraphe à partir d'une molécule
- création d'un multigraphe à partir d'un spectre de bandes
- fusion
- fission
- création d'un fichier de données pour le programme POPINF
- création d'un fichier de données pour le programme PLUTO
- représentation d'un multigraphe
- superposition de plusieurs multigraphes
- zoom d'une partie de multigraphe
- interversion de deux points d'un multigraphe
- rotation d'un multigraphe

Nous allons étudier les solutions de ces différents points en nous attardant plus particulièrement aux problèmes graphiques.

1) Point de vue logique.

a) Création d'un multigraphe à partir d'une molécule.

objectif: construire un multigraphe à partir d'un fichier contenant le numéro atomique et les coordonnées de chaque atome d'une molécule

algorithme général:

- pour chaque atome, créer un sommet
- pour chaque couple d'atomes (i,j), effectuer
 - si $\text{dist}(i-j) \leq \text{rad}(i) + \text{rad}(j) + \text{tol}$,
 - alors créer une arête entre les sommets i et j

avec $i \neq j$
 $0 < i < j$
 tol = valeur de tolérance
 $\text{dist}(i-j)$ = distance entre les atomes i et j
 $\text{rad}(i)$ = rayon de liaison de l'atome i
 $\text{rad}(j)$ = rayon de liaison de l'atome j

b) Création d'un multigraphe à partir d'un spectre de bandes.

objectif: construire un multigraphe à partir d'un fichier contenant la valeur d'énergie et la dérivée en chaque point k :

algorithme général:

- pour chaque bande du spectre, effectuer
 - créer un sommet avec le premier point
 - pour chaque point i suivant de la bande effectuer
 - créer un sommet avec le point i
 - créer une arête avec le sommet précédent

c) Fusion, fission, et création de fichiers de données pour les programmes POPINF et PLUTO.

Les solutions à ces problèmes n'étant pas compliquées, nous allons les traiter ensemble. La fusion consiste à faire une copie d'un fichier et à placer cette copie à la fin

d'un autre fichier. Entre la copie ajoutée et le contenu original, la procédure de fusion laisse de la place disponible qui pourra être utilisée ultérieurement. La fission consiste à créer un fichier dont le contenu est la copie d'une partie d'un autre fichier. Pour les deux problèmes restant, il suffit de créer des fichiers de données suivant les formats décrits dans la documentation de ces deux programmes.

d) Visualisation d'un multigraphe, zoom d'une partie de multigraphe, superposition de plusieurs multigraphes.

Les coordonnées des sommets des multigraphes sont souvent exprimées dans des unités très diverses (Å pour les molécules, point k et énergie pour les spectres, etc). Or, tous ces multigraphes doivent être représentés via un appareil graphique qui a son propre système de coordonnées. Dès lors, il faut disposer de fonctions de conversion qui permettent de passer des coordonnées d'application aux coordonnées de l'appareil.

Avant d'expliquer deux méthodes qui permettent d'accéder à ces fonctions de conversion, nous allons introduire de nouvelles notions.

•Quelques définitions:

coordonnées d'application: le système de coordonnées à deux ou à trois dimensions dans lequel les objets graphiques sont définis

ex: Å x Å x Å pour les molécules
point k x énergie pour les spectres
de bandes

coordonnées de l'appareil: il s'agit du système de coordonnées de l'appareil graphique avec lequel on travaille

ex: écran Megatek

axe X : -2048 — +2047

axe Y : -2048 — +2047

table traçante Versatec

axe X : longueur du papier en inch

axe Y : largeur du papier en inch

fenêtre (window): rectangle défini en coordonnées d'application qui permet de délimiter un objet, c'est à dire déterminer quelle partie de l'objet apparaîtra

champ de vision (viewport): rectangle défini en coordonnées d'appareil et dans lequel on désire afficher le contenu de la fenêtre

•Méthodes utilisées:

Dans le cas d'un objet bidimensionnel, on utilise les deux relations suivantes pour réaliser la conversion des coordonnées d'application en coordonnées de l'appareil graphique [6] :

$$X_{dev} = \frac{V_{xd} - V_{xg}}{W_{xd} - W_{xg}} (X_a - W_{xg}) + V_{xg} \quad (1)$$

$$Y_{dev} = \frac{V_{yh} - V_{yb}}{W_{yh} - W_{yb}} (Y_a - W_{yb}) + V_{yb} \quad (2)$$

avec (X_{dev}, Y_{dev}) = coordonnées d'un point exprimées dans le système de coordonnées de l'appareil graphique utilisé

(X_a, Y_a) = coordonnées d'un point exprimées dans le système de coordonnées de l'application

(W_{xg}, W_{yb}) = coordonnées du coin inférieur gauche de la fenêtre

(W_{xd}, W_{yh}) = coordonnées du coin supérieur droit de la fenêtre

(V_{xg}, V_{yb}) = coordonnées du coin inférieur gauche du champ de vision

(V_{xd}, V_{yh}) = coordonnées du coin supérieur droit du champ de vision

Nous allons expliquer comment obtenir la relation (1); la démonstration pourra se faire de manière analogue pour la relation (2). Sur base des définitions de la fenêtre et du champ de vision, on peut dire que les valeurs de X en coordonnées d'application varieront de W_{xg} à W_{xd} et que les valeurs de X en coordonnées d'appareil varieront de V_{xg} à V_{xd} . On va s'arranger pour que

$$\text{quand } X_a = W_{xg}, \text{ alors } X_{dev} = V_{xg} \quad (a)$$

$$\text{quand } X_a = W_{xd}, \text{ alors } X_{dev} = V_{xd} \quad (b)$$

On sait qu'une droite peut être représentée par l'équation

$$Y = mX + b \quad \text{avec } b = \text{ordonnée à l'origine} \\ m = \text{pente de la droite}$$

On remplace X par X_a et Y par X_{dev} .

$$X_{dev} = mX_a + b \quad (3)$$

Pour définir les paramètres m et b , on utilise les hypothèses (a) et (b)

$$V_{xg} = mW_{xg} + b \quad (4)$$

$$V_{xd} = mW_{xd} + b \quad (5)$$

$$(5)-(4) \quad V_{xd}-V_{xg} = mW_{xd} + b - mW_{xg} - b$$

$$m = \frac{V_{xd}-V_{xg}}{W_{xd}-W_{xg}} \quad (6)$$

$$(6) \rightarrow (4) \quad V_{xg} = \frac{V_{xd}-V_{xg}}{W_{xd}-W_{xg}} W_{xg} + b$$

$$b = V_{xg} - \frac{V_{xd}-V_{xg}}{W_{xd}-W_{xg}} W_{xg} \quad (7)$$

$$(6), (7) \rightarrow (3) \quad X_{dev} = \frac{V_{xd}-V_{xg}}{W_{xd}-W_{xg}} X_a + V_{xg} - \frac{V_{xd}-V_{xg}}{W_{xd}-W_{xg}} W_{xg}$$

$$X_{dev} = \frac{V_{xd}-V_{xg}}{W_{xd}-W_{xg}} (X_a - W_{xg}) + V_{xg} \quad (1)$$

Ces relations permettent de réaliser un zoom d'une partie d'un objet représenté. En effet, pour obtenir un agrandissement, il suffit de diminuer la fenêtre sans toucher au champ de vision, c'est à dire qu'il faut modifier les paramètres W_{xg} , W_{xd} , W_{yb} , W_{yh} .

D'autre part, pour réaliser la superposition de multigraphes, il suffit d'utiliser les mêmes fonctions de conversion pour représenter les différents multigraphes.

Dans le cas d'objets tridimensionnels, on utilise les 3 relations suivantes:

$$X_{\text{dev}} = [(X_{\text{slope}} * X_a) + X_{\text{intcp}}] * VSX + VCX$$

$$Y_{\text{dev}} = [(Y_{\text{slope}} * Y_a) + Y_{\text{intcp}}] * VSY + VCY$$

$$Z_{\text{dev}} = [(Z_{\text{slope}} * Z_a) + Z_{\text{intcp}}] * VSZ + VCZ$$

avec (X_a, Y_a, Z_a) = coordonnées d'un point exprimées dans le système de coordonnées de l'application

$(X_{\text{dev}}, Y_{\text{dev}}, Z_{\text{dev}})$ = coordonnées d'un point exprimées dans le système de coordonnées de l'appareil graphique utilisé

(VCX, VCY, VCZ) = coordonnées du point central du champ de vision

VSX = demi-largeur du champ de vision

VSY = demi-hauteur du champ de vision

VSZ = demi-profondeur du champ de vision

$X_{\text{slope}}, X_{\text{intcp}}$ = pente et ordonnée à l'origine de la droite de conversion pour la coordonnée X

$Y_{\text{slope}}, Y_{\text{intcp}}$ = idem pour la coordonnée Y

$Z_{\text{slope}}, Z_{\text{intcp}}$ = idem pour la coordonnée Z

On utilise un cube comme champ de vision afin de ne pas distordre les objets graphiques.

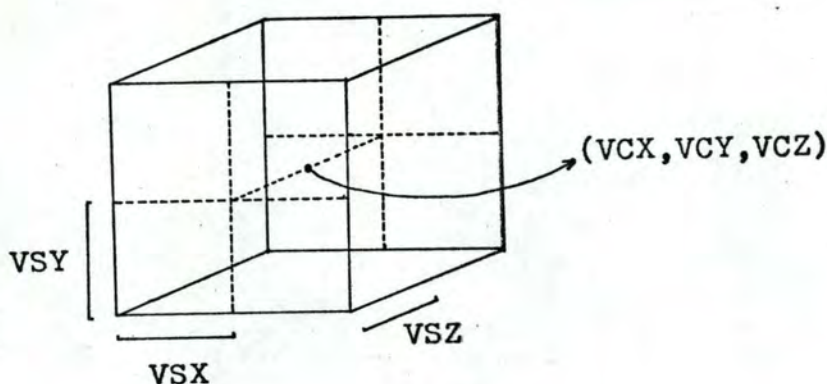


Fig. VI.1

Nous allons expliquer la manière de définir les paramètres Xslope et Xintcp de la relation

$$X_{dev} = [(Xslope * \dot{X}_a) + Xintcp] * VSX + VCX \quad (1)$$

$$\text{On pose } A = (Xslope * X_a) + Xintcp \quad (2)$$

De manière à ce que le paramètre A varie de -1 à +1,
on pose $A = 1$ quand $X_a = X_{max}$
 $A = -1$ quand $X_a = X_{min}$

En introduisant ces hypothèses dans la relation (2),
on obtiendra

$$1 = Xslope * X_{max} + Xintcp \quad (3)$$

$$-1 = Xslope * X_{min} + Xintcp \quad (4)$$

$$(3)-(4) \rightarrow 2 = Xslope * X_{max} + \cancel{Xintcp} - Xslope * X_{min} - \cancel{Xintcp}$$

$$Xslope = \frac{2}{X_{max} - X_{min}} \quad (5)$$

$$(5) \rightarrow (3) \quad 1 = \frac{2}{X_{\max} - X_{\min}} * X_{\max} + X_{\text{intcp}}$$

$$X_{\text{intcp}} = 1 - \frac{2 * X_{\max}}{X_{\max} - X_{\min}} \quad (6)$$

$$(5), (6) \rightarrow (2)$$

$$A = \frac{2}{X_{\max} - X_{\min}} * X_a + 1 - \frac{2 * X_{\max}}{X_{\max} - X_{\min}}$$

$$\text{si } X_a = X_{\max} \quad \text{alors } A = 1$$

$$\begin{aligned} X_a = X_{\min} \quad \text{alors } A &= \frac{2 X_{\min}}{X_{\max} - X_{\min}} + 1 - \frac{2 X_{\max}}{X_{\max} - X_{\min}} \\ &= -2 \frac{X_{\max} - X_{\min}}{X_{\max} - X_{\min}} + 1 \\ &= -1 \end{aligned}$$

Lorsque l'implémentation sera abordée, on expliquera la raison pour laquelle une méthode a été utilisée pour les objets bidimensionnels et une autre pour les objets tridimensionnels.

e) Interversion de deux points d'un multigraphe et rotation de multigraphe.

La solution de chacun de ces problèmes n'est pas générale mais a été réalisée en fonction des appareils graphiques disponibles (joystick, HCRST). Ces solutions seront explicitées plus loin quand on parlera de l'implémentation.

2) Point de vue implémentation.

a) 3 programmes.

Les solutions aux six premiers problèmes ont donné lieu à autant de sousroutines qui ont été rassemblées en un seul programme (FICPRO) vu que toutes produisaient un fichier pour résultat.

Deux programmes de visualisation ont été implémentés; le premier travaille avec le Megatek et le logiciel graphique WAND, le second travaille avec la table traçante Versatec et le logiciel VERSAPLOT.

En fait, pour réaliser le second programme de visualisation, il a suffi de prendre les routines de dessin déjà existantes et de remplacer les routines graphiques du package WAND par celles de VERSAPLOT.

b) Overlay.

Avant de voir en détail le premier programme graphique, une remarque s'impose. On sait que la taille de la mémoire du PDP 11/60 n'est pas énorme; dès lors, afin de pouvoir malgré tout développer des programmes importants, on a recours à la technique du recouvrement (OVERLAY). De quoi s'agit-il ?

En fait, dans un long programme d'application, on utilise rarement tous les modules en même temps, mais seulement quelques-uns parmi l'ensemble. Il est donc inutile de charger tout en mémoire centrale. Avec la technique du recouvrement, seuls les modules nécessaires sont chargés.

Pratiquement, cela se passe de la manière suivante. Lors du link du programme, il faut définir à l'aide du langage ODL (Overlay Description Language) une arborescence qui représente une structure de recouvrement. Dans cette structure, on distingue des modules permanents, ceux qui appartiennent à la racine de l'arborescence, et des modules rechargeables, ceux qui appartiennent à une des branches.

Les modules permanents seront chargés en mémoire centrale pendant toute l'exécution du programme. Quand un module rechargeable devra être exécuté, la branche de l'arborescence à laquelle il appartient sera chargée.

Pendant tout le développement du programme, il a fallu porter attention à ce problème de structure de recouvrement; c'est pourquoi on a préféré développer beaucoup de courtes sous-routines qui réalisent une et une seule action bien précise plutôt que de longues procédures qui peuvent réaliser plusieurs actions.

NB: en annexe, sont reprises les structures d'overlay des 3 programmes.

c) Description du programme graphique DESSIN.

Le graphe de la figure VI.2 donne la structure physique du programme. Les sommets représentent les sous-routines et les arêtes, les appels entre sous-routines.

On peut classer les différentes sous-routines en six catégories en fonction de leur rôle: initialisation, utilitaire, présentation de menu, préparation à la représentation, représentation, modification.

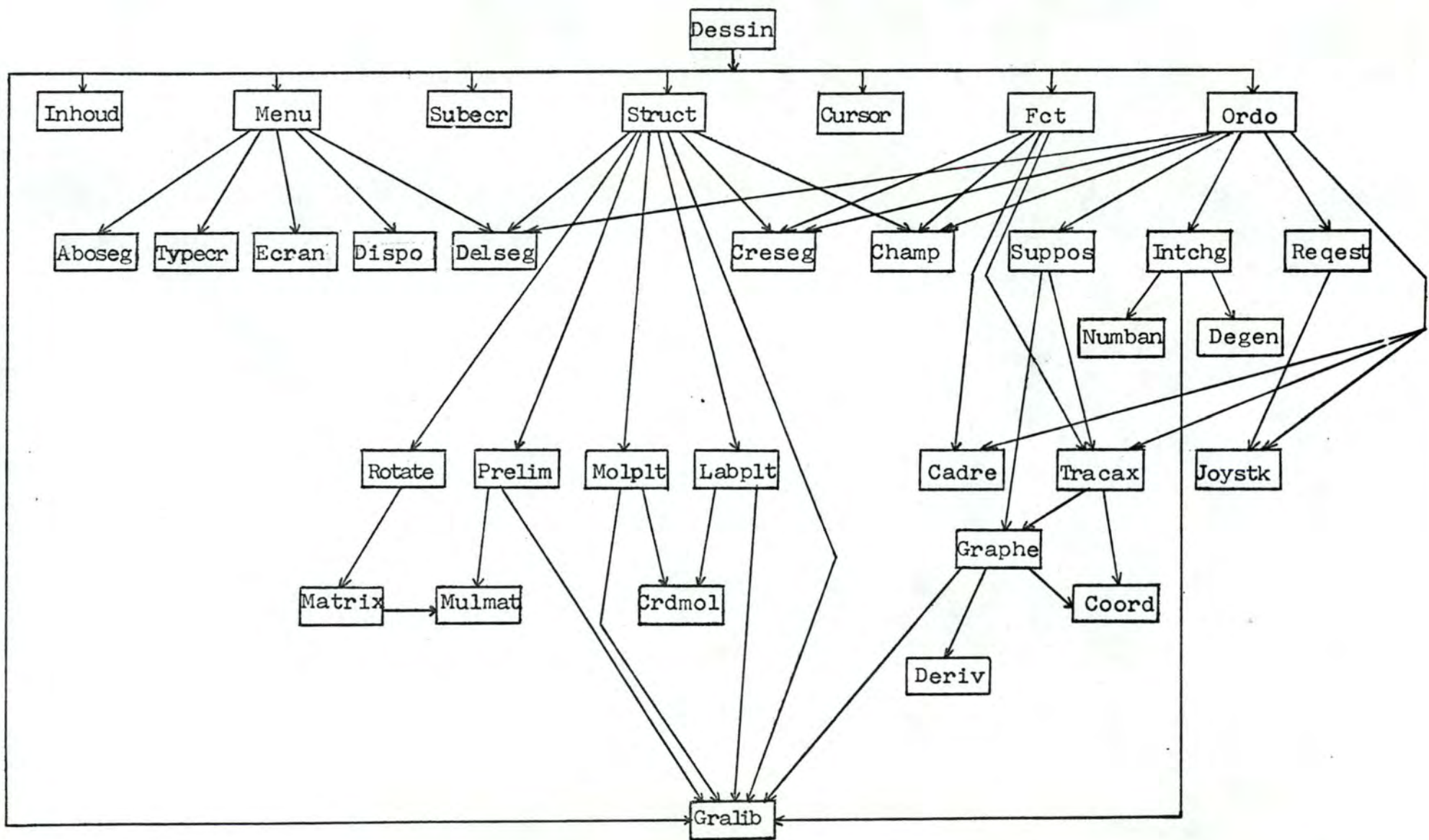


Fig. VI.2 : Structure physique du programme DESSIN.

1° classe de routines = initialisation

routines: debut, inhou, subecr, cursor.

Dans la partie d'initialisation du programme, on trouve l'ouverture des fichiers de travail (DEBUT), la définition d'objets graphiques qui seront utilisés dans le programme (curseurs, division de l'écran) et surtout la routine INHOUD qui joue un rôle capital pour le déroulement du programme.

On sait que, grâce à la procédure de fusion, l'utilisateur a la possibilité de stocker la description de plusieurs multigraphes dans les deux fichiers de travail. Il nous faut donc une routine qui fera un relevé des différents multigraphes décrits dans les fichiers. Pour chaque multigraphe, on enregistre dans une table les paramètres suivants:

- le numéro du premier record dans le fichier des sommets
- le numéro du dernier record dans le fichier des sommets
- le nombre de sommets
- le numéro du premier record dans le fichier des arêtes
- le numéro du dernier record dans le fichier des arêtes
- le nombre d'arêtes
- le titre.

Ces paramètres sont très importants car ils permettent de localiser parfaitement la description d'un multigraphe spécifique dans les deux fichiers. Ainsi, quand l'utilisateur désirera travailler avec un multigraphe particulier, on pourra à l'aide de ces paramètres accéder immédiatement aux informations.

D'autre part, ces paramètres ne sont jamais modifiés lors de l'exécution du programme graphique parce que dans les applications étudiées, l'utilisateur n'ajoute ni ne supprime d'éléments au multigraphe. Il est certain que dans un contexte dynamique (c'est à dire avec ajout et/ou suppression), la table des paramètres devrait être mise à jour après chaque modification.

2° classe de routines = "utilitaires"

routines: aboseg, delseg, creseg, champ, joystk, mulmat.

On a regroupé sous le terme "utilitaire" les routines qui sont utilisées quel que soit le type d'application.

Avant de décrire le fonctionnement des différentes routines, nous allons introduire une nouvelle définition: on appelle segment graphique une collection d'instructions représentant des primitives graphiques que l'on peut manipuler comme une unité.

CRESEG, DELSEG, ABOSEG

Dans le logiciel WAND, tout segment graphique est identifié par un numéro. Un segment nouvellement créé ne peut donc pas porter un numéro déjà attribué à un autre segment existant car cela provoquerait une erreur fatale à l'exécution.

D'autre part, le programme offre à l'utilisateur la possibilité de diviser l'écran graphique en plusieurs parties et de choisir la partie de l'écran où le multigraphe doit être visualisé.

Les différentes possibilités de division de l'écran sont:

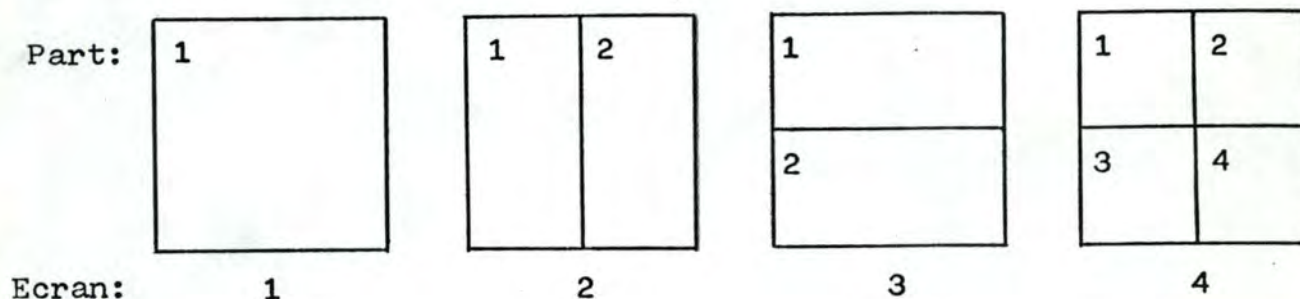


Fig. VI.3

Le calcul du numéro de segment est réalisé à l'aide de la relation

$$\text{Numéro de segment} = \text{Ecran} \times 10 + \text{Part}$$

Les seuls numéros possibles de segment sont 11, 21, 22, 31, 32, 41, 42, 43, 44, et au maximum quatre multigraphes peuvent être visualisés sur l'écran graphique. Les numéros des segments existants seront donc stockés dans un tableau à quatre cases.

Après cette introduction, nous pouvons revenir au rôle de chacune des 3 routines.

CRESEG: cette routine a pour rôle de calculer le numéro du segment qui sera créé et de stocker ce numéro dans la table.

DELSEG: cette routine a pour rôle de vérifier s'il n'existe pas déjà un segment identifié par un numéro X. Si tel est le cas, on supprime le segment et on enlève le numéro de la table.

ABOSEG: cette routine a pour rôle de supprimer tous les segments affichés à l'écran et de remettre à zéro toutes les cases de la table.

Les autres routines utilitaires sont:

CHAMP: cette routine a pour rôle de définir les paramètres du champ de vision choisi par l'utilisateur. Les différents champs de vision sont décrits à la figure précédente.

JOYSTK: cette routine a pour rôle de renvoyer les coordonnées d'écran d'un point sélectionné à l'aide d'un curseur déplacé par le joystick.

MULMAT: cette routine a pour rôle de réaliser la multiplication de deux matrices (c'est à dire des tableaux bidimensionnels).

3° classe de routines = présentation de menu
.....

routines: dessin, menu, fct, ordo, struct, typecr,
dispo, ecran.

Nous n'allons pas nous attarder sur ces routines car elles ne présentent pas grand intérêt au niveau de l'algorithme. Il s'agit de routines qui affichent un menu au vidéo, acceptent et valident le choix de l'utilisateur, et orientent la suite du traitement.

4° classe de routines = préparation à la représentation
.....

routines: cadre, coord, prelim, crdmol.

La routine CADRE calcule les paramètres des relations de conversion pour les multigraphes bidimensionnels, tandis que la routine PRELIM réalise ces mêmes calculs pour les multigraphes tridimensionnels.

Nous allons expliquer pourquoi deux méthodes de conversion des coordonnées ont été implémentées. Avec les objets graphiques tridimensionnels, apparaît un problème nouveau: il faut absolument respecter la géométrie de l'objet afin que l'image ne soit pas distordue.

Ainsi, lorsqu'un cycle benzénique doit être représenté, il est indispensable d'obtenir un hexagone régulier et non pas un hexagone étiré; de plus, il faut aussi qu'une arête représentant une liaison C-C (longueur=1.54 Å) soit plus grande qu'une arête représentant une liaison C-H (longueur=1.09 Å).

Pour ne pas générer d'image distordue, les deux précautions suivantes doivent être prises:

- le champ de vision sera un cube, c'est à dire que les fourchettes de variation des 3 axes sont identiques en valeur absolue,
- on s'arrangera pour qu'au niveau des coordonnées d'application, les trois coordonnées aient aussi la même fourchette de variation en valeur absolue.

La conséquence de la première précaution est de ne pas utiliser toute la place disponible de la partie de l'écran où l'objet doit être visualisé. Or, avec un graphique à deux dimensions, cela a peu d'importance d'avoir un multi-

graphe étiré, au contraire cela produit un effet d'agrandissement suivant l'un des axes. C'est la raison qui nous a poussés à implémenter deux méthodes de conversion de coordonnées.

5° classe de routines = représentation
.....

routines: tracax, graphe, molplt, labplt, suppos, deriv.

Quand les fonctions de conversion ont été calculées, on peut représenter le multigraphe à l'écran graphique. Ceci est réalisé par différentes routines:

TRACAX: représente le système d'axes (X,Y) avec titres des axes.

GRAPHE: représente un multigraphe bidimensionnel avec des traits.

MOLPLT: représente un multigraphe tridimensionnel avec des traits.

LABPLT: représente un multigraphe tridimensionnel avec des traits et des labels.

SUPPOS: représente plusieurs multigraphes bidimensionnels.

DERIV: représente la dérivée première par un vecteur rotatif en chaque point du spectre de bandes.

L'algorithme général pour les routines graphe, molplt et labplt consiste, pour chaque arête du multigraphe, à aller chercher les coordonnées des sommets, appliquer les relations de conversion et tracer un trait entre les deux points.

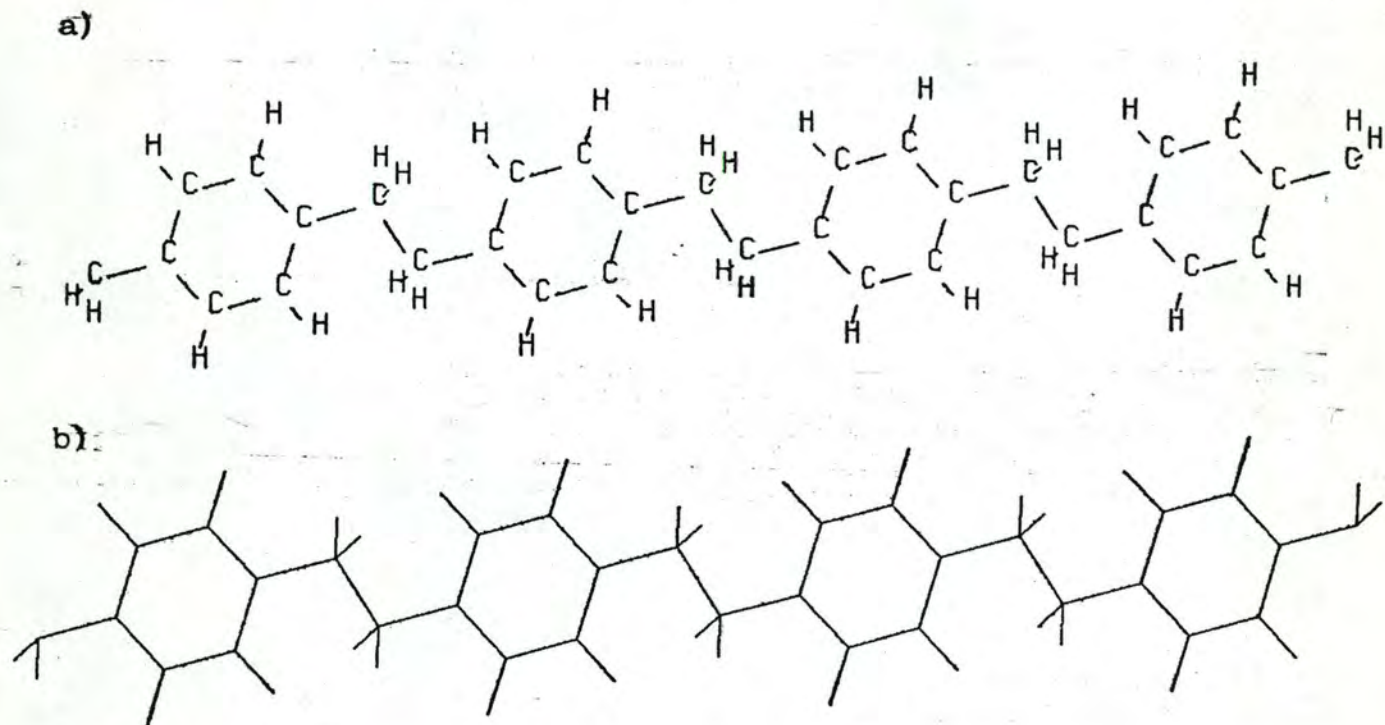


Fig. VI.4 : Représentation a) avec symboles atomiques, b) sans symbole atomique du polyparaxylylène.

6° classe de routines = modification
.....

routines: intchg, numban, degen, re quest, rotate, matrix.

Ces fonctions de modification doivent résoudre les deux problèmes laissés en suspens lors du paragraphe précédent. Il s'agit de la rotation des molécules et de l'interversion de deux points du spectre de bandes.

La procédure de rotation s'effectue suivant l'algorithme général suivant:

- 1) acquérir la valeur des angles de rotation à l'aide de la boîte de fonction [ROTATE]
- 2) construire la matrice de transformation [MATRIX]

- 3) appliquer la matrice de transformation au segment graphique. Ceci est réalisé par hardware [ROTATE]
- 4) si arrêt = oui, arrêter la procédure sinon, retourner en 1 .

La procédure pour intervertir deux points du spectre de bandes est la suivante:

- 1) acquérir les coordonnées d'écran des deux points à intervertir [REQUEST]
- 2) calcul de la valeur du point k de ces deux points [INTCHG]
- 3) proposition de numéros pour la bande inférieure et la bande supérieure [DEGEN, NUMBAN].
- 4) si confirmation à la proposition, modifier les données dans les fichiers de travail [INTCHG].

Deux algorithmes sont utilisés pour la proposition des numéros de bandes. Si les points sélectionnés ne sont pas confondus avec d'autres points du spectre, le programme utilise un algorithme qui fait une seule proposition de numéros de bandes; par contre, si les points sélectionnés sont confondus avec d'autres points du spectre (cas des sommets finals d'une bande), un tel algorithme ne sait pas trouver directement la solution.

exemple:

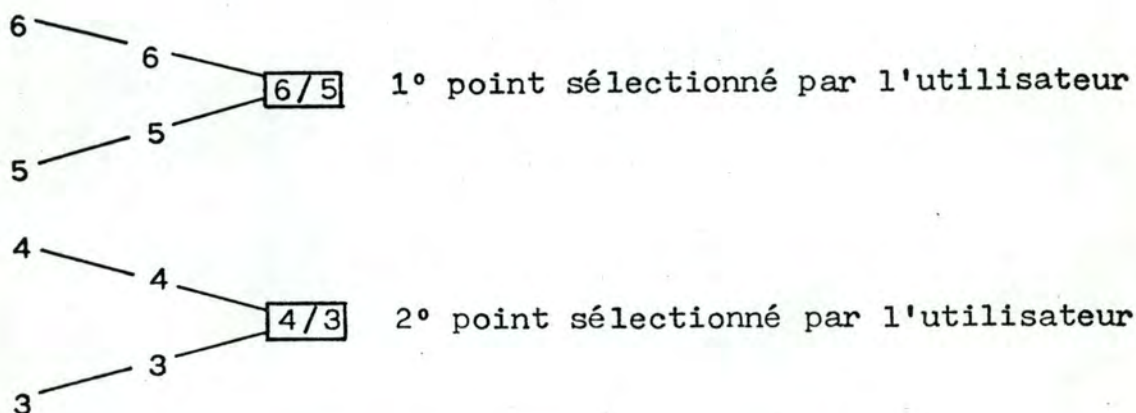


Fig. VI.5

Les différentes possibilités sont: intervertir 3 avec 5
 3 avec 6
 4 avec 5
 4 avec 6

Un algorithme qui fait une seule proposition n'a aucun moyen de trouver le bon couple de valeurs; c'est pourquoi pour de tels points, on a préféré implémenter un algorithme qui propose toutes les possibilités et qui attend une confirmation de la part de l'utilisateur.

d) Description du programme graphique HRDCPY.

Pour implémenter le programme HRDCPY, les routines du logiciel WAND ont été remplacées par les routines du logiciel VERSAPLOT. Nous n'allons donc pas nous attarder sur ce programme. Il faut cependant faire remarquer qu'une option

supplémentaire a été développée pour le dessin sur papier: il s'agit de la représentation en stéréoscopie.

La stéréoscopie a pour but de fournir à l'observateur la perception de la profondeur à partir d'images planes, en présentant à chacun des deux yeux une image légèrement différente du champ d'observation.

Pratiquement, une vue stéréoscopique est réalisée de la manière suivante; on applique une rotation de -3° (autour de l'axe vertical du dessin) par rapport à l'objet original et cela donne le dessin de gauche, puis on applique une rotation de $+3^\circ$ par rapport à l'objet original et cela donne le dessin de droite. Finalement, les deux dessins doivent être rapprochés de sorte que deux points identiques soient séparés par une distance d'environ 60 mm (ce qui correspond à la distance entre les deux yeux). L'effet stéréoscopique peut être perçu à l'aide de lunettes spéciales.

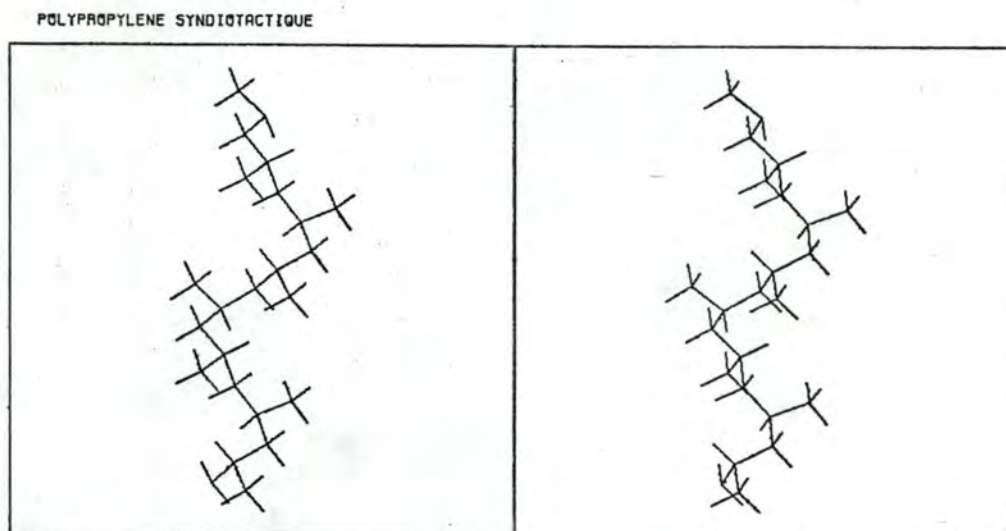


Fig. VI.6

CHAPITRE VII

Manuel d'utilisation.

1) Programme FICPRO.

Dans ce programme ont été rassemblées toutes les routines qui créent des fichiers. Six options s'y trouvent.

a) Option 1: création d'un multigraphe à partir d'une molécule.

Rôle: définir l'ensemble des liaisons entre les atomes de la structure sur base des coordonnées cartésiennes de chaque atome et créer un multigraphe représentant cette structure.

Schéma:

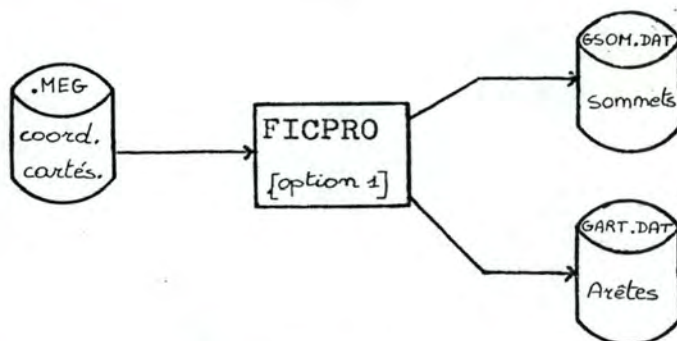


Fig. VII.1

Entrée: fichier avec coordonnées cartésiennes de chaque atome.

Sortie: fichiers contenant la description des sommets et la description des arêtes du multigraphe.

Utilisation du programme: après avoir choisi l'option "Création de multigraphe à partir d'une molécule", l'utilisateur doit spécifier le nom de son fichier d'entrée et un titre pour identifier le multigraphe qui sera créé.

b) Option 2: création d'un multigraphe à partir d'un spectre.

Rôle: définir un multigraphe représentant un spectre de bandes.

Schéma:

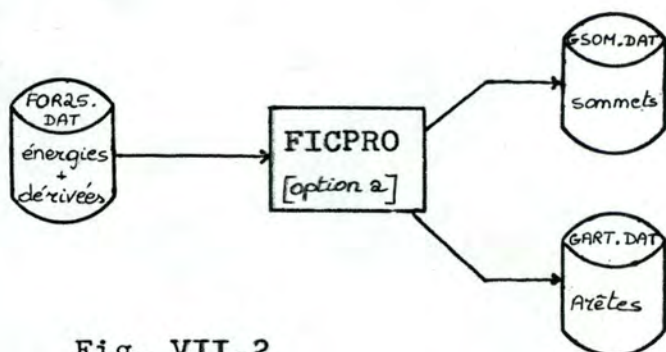


Fig. VII.2

Entrée: fichier avec les valeurs des énergies et des dérivées premières du spectre de bandes.

Sortie: fichiers contenant la description des sommets et la description des arêtes du multigraphe.

Utilisation du programme: après avoir choisi l'option "Création de multigraphe à partir d'un spectre", l'utilisateur doit spécifier:

- un titre pour identifier le multigraphe qui sera créé
- le nombre total de bandes dans le spectre
- le nombre de points k
- le nom du fichier de données
- s'il désire un multigraphe représentant tout le spectre ou seulement une partie. Dans le second cas, l'utilisateur devra spécifier le numéro de la première bande et de la dernière bande du spectre dont il désire faire un multigraphe.

c) Option_3: fusion_de_fichiers.

Rôle: fusionner plusieurs fichiers contenant chacun la description des sommets d'un multigraphe en un seul fichier contenant la description des sommets de tous les multigraphes;

fusionner plusieurs fichiers contenant chacun la description des arêtes d'un multigraphe en un seul fichier contenant la description des arêtes de tous les multigraphes.

Remarque: si on désire travailler sur un seul multigraphe avec le programme DESSIN, cette étape est inutile.

Schéma:

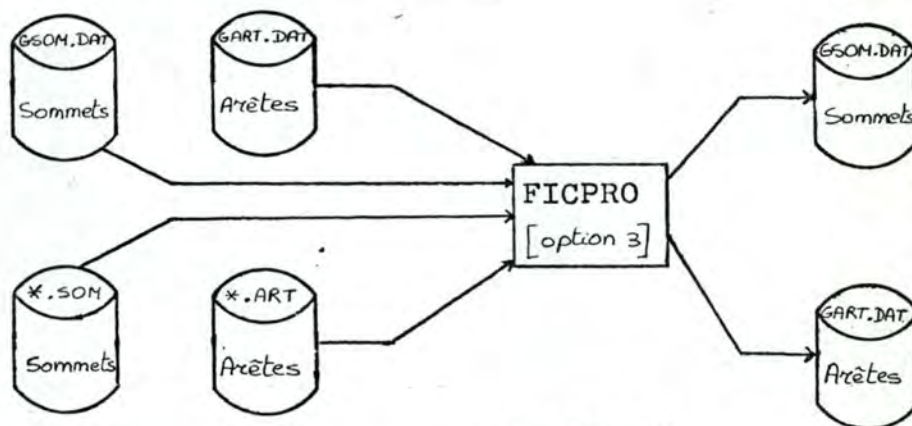


Fig. VII.3

Entrée: deux fichiers contenant la description des sommets et deux fichiers contenant la description des arêtes.

Sortie: un fichier contenant la description des sommets des deux multigraphes et un fichier contenant la description des arêtes des deux multigraphes.

Utilisation du programme: après avoir choisi l'option "Fusion de fichiers", l'utilisateur doit spécifier:

le nom du fichier avec la description des sommets
le nom du fichier avec la description des arêtes.

NB: cette procédure est à répéter pour chaque multigraphe que l'on désire fusionner.

d) Option 4: fission de fichiers.

Rôle: à partir d'un fichier contenant la description des sommets de plusieurs multigraphes, créer un fichier qui contiendra une copie de la description des sommets d'un de ces multigraphes;

à partir d'un fichier contenant la description des arêtes de plusieurs multigraphes, créer un fichier qui contiendra une copie de la description des arêtes d'un de ces multigraphes.

Schéma:

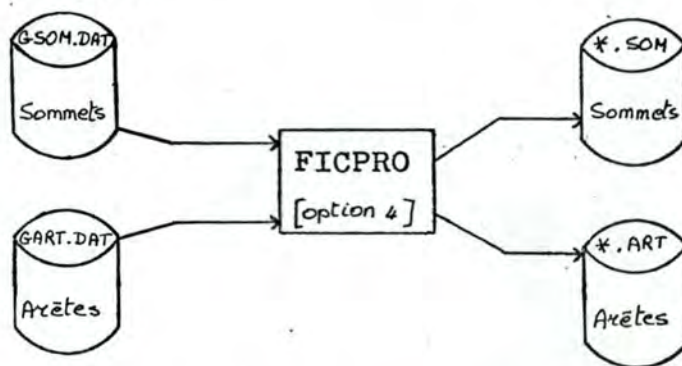


Fig. VII.4

Entrée: un fichier contenant la description des sommets de plusieurs multigraphes et un fichier contenant la description des arêtes de plusieurs multigraphes.

Sortie: un fichier contenant la description des sommets et un fichier contenant la description des arêtes du multigraphe sélectionné.

Utilisation du programme: après avoir choisi l'option "Fission de fichier", le programme affiche le titre des multigraphes disponibles dans les fichiers de données. L'utilisateur doit spécifier:

- le numéro du multigraphe dont il désire faire une copie
- le nom du fichier avec la description des sommets
- le nom du fichier avec la description des arêtes.

e) Option_5: fichier de données pour POPINF.

Rôle: créer un fichier de données pour le programme POPINF.

Schéma:

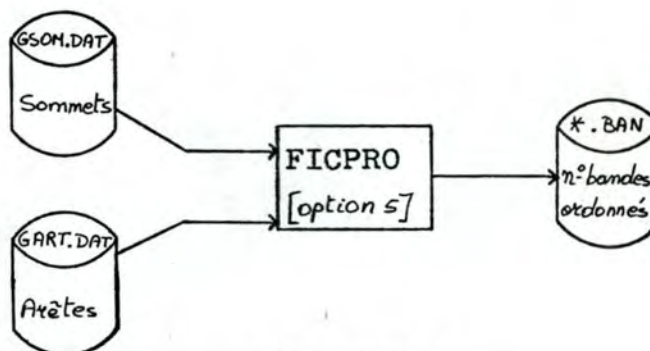


Fig. VII.5

Entrée: fichiers contenant la description des sommets et la description des arêtes de multigraphes.

Sortie: fichier contenant l'ensemble des numéros de bandes ordonnés

Utilisation du programme: après avoir choisi l'option "Fichier de données pour POPINF", le programme affiche le titre des multigraphes disponibles dans les fichiers de données. L'utilisateur doit spécifier:

- le numéro du multigraphe avec lequel il désire travailler
- le nombre de points k
- le nombre de bandes
- le nom du fichier de sortie.

f) Option 6: fichier de données pour PLUTO.

Rôle: créer un fichier de données et un fichier de commande pour le programme PLUTO.

Schéma:

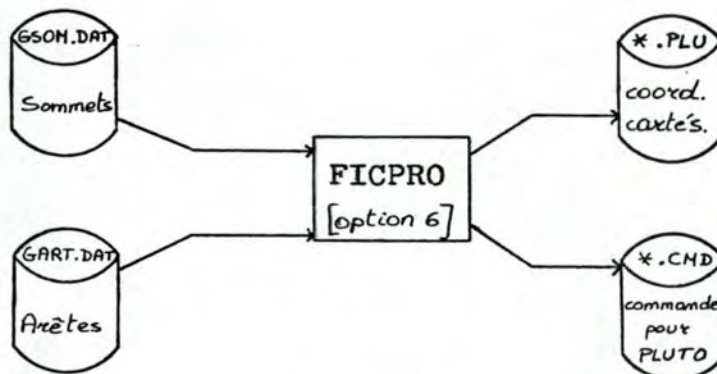


Fig. VII.6

Entrée: un fichier contenant la description des sommets de plusieurs multigraphes et un fichier contenant la description des arêtes de plusieurs multigraphes.

Sortie: fichier de données qui, pour chaque atome de la structure, reprend le symbole atomique, le numéro de l'atome et les coordonnées cartésiennes;

fichier de commande qui contient les ordres du programme Pluto en fonction des choix de l'utilisateur.

Utilisation du programme: après avoir choisi l'option "Fichier de données pour Pluto", le programme affiche le titre des multigraphes disponibles dans les fichiers de données. L'utilisateur doit spécifier:

- le numéro du multigraphe avec lequel il désire travailler
- le nom du fichier de données
- le nom du fichier de commande,

et il devra choisir une option parmi les menus suivants:

type de représentation

- a) avec tiges
avec tiges et boules
avec sphères de Van der Waals
- b) mono
stéréo
- c) avec ombrage
sans ombrage
- d) aucun label
tous les labels
certains labels.

2) Programme DESSIN.

Rôle: représenter à l'écran un multigraphe et donner à l'utilisateur la possibilité de le modifier.

Schéma:

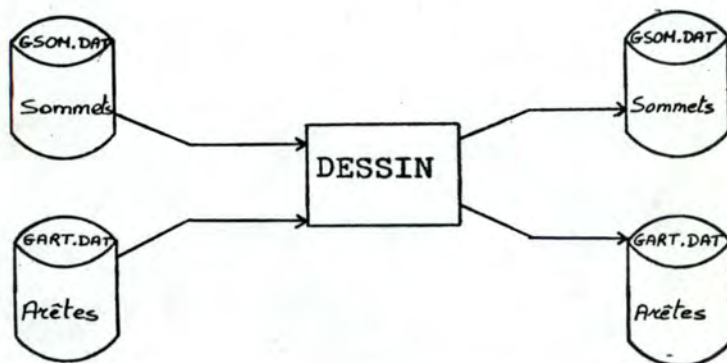


Fig. VII.7

Entrée: fichiers contenant la description des sommets et la description des arêtes de multigraphes.

Sortie: fichiers contenant la mise à jour de la description des sommets et des arêtes des multigraphes.

Utilisation du programme: de manière générale, le programme fonctionne de la façon suivante:

- présentation au vidéo d'un menu principal avec différentes options
- l'utilisateur tape le chiffre correspondant à son choix, puis la touche <Return>
- présentation détaillée de l'option choisie
- nouveau choix de l'utilisateur
- puis un dialogue (questions-réponses) s'engage entre le programme et l'utilisateur.

Les différentes options du menu principal sont:

- 1: Division de l'écran
- 2: Choix du multigraphe
- 3: Choix du type de dessin
- 4: Arrêt du programme

Description des différentes options:

- division de l'écran: cette option permet à l'utilisateur de diviser la surface disponible de l'écran graphique en plusieurs parties; les différentes possibilités sont:

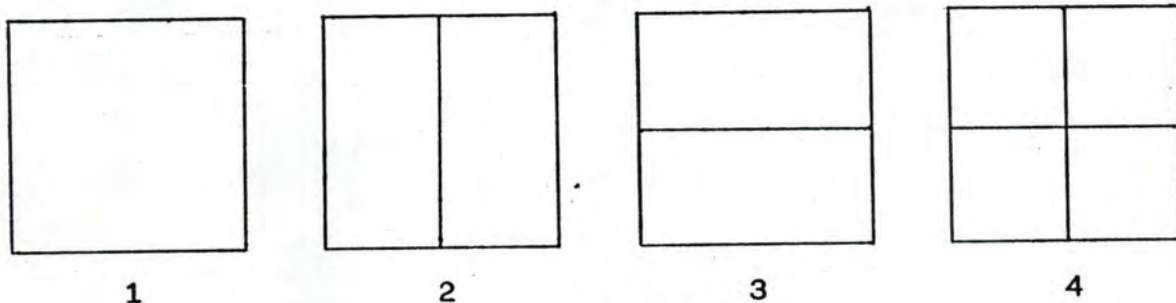


Fig. VII.8

- choix du multigraphe: le programme affiche au vidéo le titre de tous les multigraphes disponibles dans les fichiers de données et l'utilisateur choisit le multigraphe sur lequel il désire travailler;
- choix du type de dessin: les options disponibles actuellement sont:

"1: molécule ou polymère" , avec pour menu secondaire:

1: Représentation de la structure

-sans symboles chimiques

-avec symboles chimiques

-avec atomes d'hydrogène

-sans atome d'hydrogène

2: Rotation

"2: ordonnancement" , avec pour menu secondaire:

1: Type de représentation

-avec arêtes

-avec labels

2: Agrandissement d'une zone

3: Choix des points à intervertir

4: Coordonnées d'un point

5: Superposition de multigraphes

"3: représentation de fonction"

Explication des différentes options:

•Pour l'option 'molécule ou polymère'

-l'utilisateur est obligé de représenter la molécule avant d'appliquer des rotations

-le système d'axes utilisé pour les rotations est le système gaucher suivant:

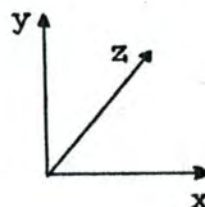


Fig. VII.9

-pour réaliser les rotations, il faut utiliser les 4 boutons supérieurs de la boîte de fonction

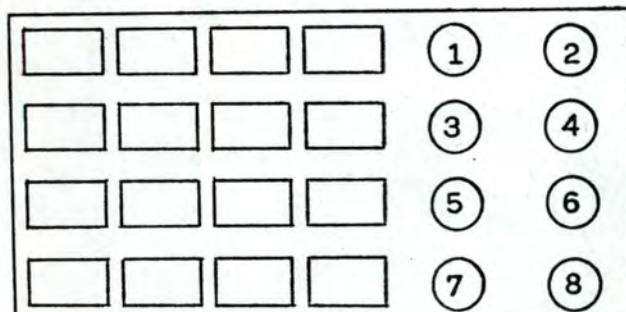


Fig. VII.10

le bouton ① pour la rotation autour de l'axe X
 le bouton ② pour la rotation autour de l'axe Y
 le bouton ③ pour la rotation autour de l'axe Z
 le bouton ④ pour revenir au menu "molécule ou polymère"

au début, les boutons doivent être tournés dans le sens horlogique

•Pour l'option 'ordonnancement'

-l'utilisateur doit d'abord représenter le spectre

-pour sélectionner les points à intervertir, l'utilisateur déplace le curseur à l'aide du joystick afin de l'amener sur le numéro du point à choisir; quand le curseur est positionné correctement, il faut appuyer sur le bouton du joystick;

-quand l'utilisateur a confirmé l'intervention de deux points, une croix apparaîtra à gauche des numéros des deux points sensibilisés.

3) Programme HRDCPY.

Rôle: représenter sur papier un multigraphe.

Schéma:

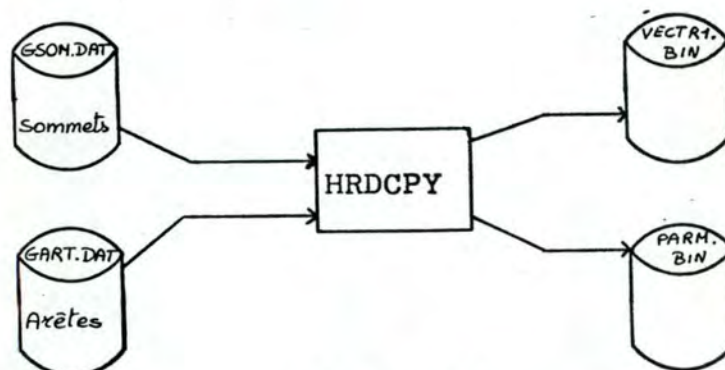


Fig. VII.11

Entrée: fichiers contenant la description des sommets et la description des arêtes des multigraphes.

Sortie: deux fichiers binaires contenant les paramètres indispensables pour la représentation sur papier.

Utilisation du programme: l'utilisateur doit d'abord spécifier la longueur maximale (Xmax) de papier que l'imprimante pourra utiliser pour représenter les dessins créés par le programme HRDCPY.

Ce paramètre se calcule de la manière suivante:

$$X_{max} = 11 \times \text{nombre de dessins que l'utilisateur désire créer pendant une exécution du programme HRDCPY.}$$

exemple: si 10 dessins à créer,

VP7>XMAX=110 <Return>

VP7>/E

(instruction qui termine
l'initialisation et lance
l'exécution)

affiché par le programme à introduire par l'utilisateur

Puis, le programme affiche le titre de tous les multigraphes disponibles dans les fichiers de données, l'utilisateur choisit celui avec lequel il désire travailler et définit le type de dessin à l'aide des options proposées, c'est à dire:

Molécule ou polymère

Mono

Séréo

Avec symboles atomiques

sans atome d'hydrogène

avec atomes d'hydrogène

Taille du cadre

20 cm

15 cm

10 cm

Fonction

Dans cette option, l'utilisateur devra spécifier les paramètres suivants:

Xmax = valeur maximale de l'axe X

Xmin = valeur minimale de l'axe X

Ymax = valeur maximale de l'axe Y

Ymin = valeur minimale de l'axe Y

(Xo,Yo) = coordonnées du point d'intersection
des deux axes

titre pour axe X

titre pour axe Y

CHAPITRE VIII

Evaluation et perspectives.

Dans ce dernier chapitre, nous aborderons les limitations de ce travail.

1) Limite de la généralisation des problèmes graphiques.

Le principe qui consiste à transformer un objet graphique en multigraphe est spécialement bien adapté aux dessins rencontrés dans les problèmes scientifiques. En effet, la majorité de ces dessins sont définis dans un système de coordonnées, ce qui facilite le passage à la visualisation sur écran graphique ou sur papier. Par contre, les programmes de ce travail ne sont pas à même de représenter des diagrammes de flux, des organigrammes, des schémas d'accès, ..., c'est à dire des dessins qui ne sont pas définis naturellement dans un système de coordonnées.

2) Librairie GRALIB.

Pour l'instant, cette librairie ne contient pas de primitives de suppression d'éléments du multigraphe. D'un autre côté, l'utilisateur n'a pas la possibilité d'ajouter des sommets ou arêtes à un multigraphe déjà défini, il peut seulement modifier les informations d'éléments existants. Ce contexte assez statique dans lequel on travaille, n'est pas une contrainte pour les deux applications étudiées dans le mémoire. En effet, il n'y a aucun intérêt à supprimer un atome ou une liaison dans une molécule.

Néanmoins, il serait intéressant d'envisager des applications où l'utilisateur aurait la possibilité d'ajouter, de supprimer des éléments du multigraphe de manière interactive

et de visualiser directement la modification à l'écran graphique. Pour permettre ce type d'application, il faut non seulement implémenter des primitives de suppression, mais aussi gérer l'emplacement des enregistrements dans les fichiers de travail qui peuvent contenir la description de plusieurs multigraphes.

On voit donc que de nouveaux problèmes surgissent quand on passe d'applications assez limitées au niveau des modifications, à des applications dynamiques.

3) Programmes de visualisation.

Les deux programmes graphiques implémentés permettent de résoudre les problèmes qui étaient posés au début de ce travail et constituent un outil de base en ce sens qu'on peut représenter n'importe quel objet graphique défini dans un système de coordonnées et décrit sous la forme d'un multigraphe.

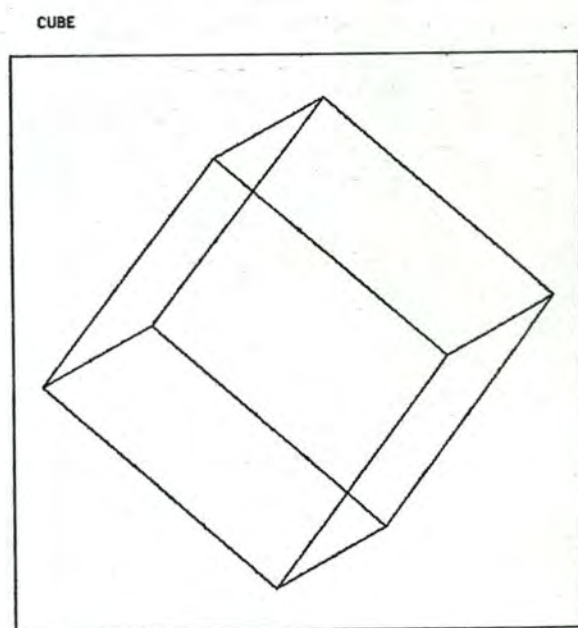
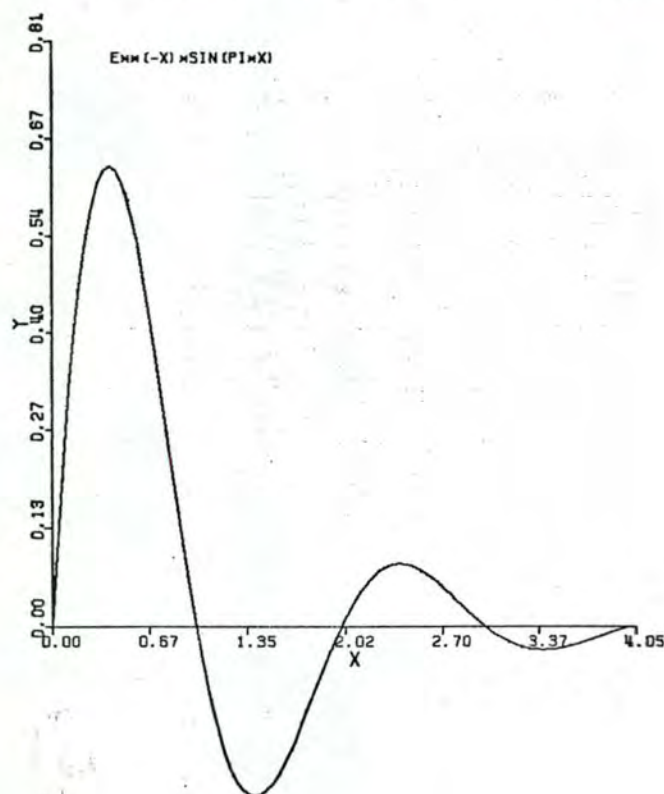


Fig. VIII.1

Naturellement, des améliorations peuvent être apportées en fonction d'applications spécifiques; par exemple la superposition de molécules, la vue stéréoscopique à l'écran graphique, la description d'un multigraphe de manière interactive à l'aide de la tablette et de la souris avec visualisation à l'écran,

En conclusion, il faut constater qu'un tel travail est loin d'être terminé et qu'avec de l'imagination, de nouvelles possibilités graphiques peuvent être envisagées pour des travaux ultérieurs.

Bibliographie

et

Annexe.

Bibliographie.

- [1] Introduction to WAND 7200 - Mai 1981
- [2] VERSAPLOT Operating Manual PDP-11 RSX-11M
- [3] MEGATEK Whizzard 7000 User's Manual
- [4] VAN WONTERGHEM V.
Mémoire de licence en sciences chimiques-FNDP-(1981)
- [5] MOTHERWELL S.
Pluto - User's Manual (Version 1)
- [6] NEWMAN W.M., Sproull R.F.
Principles of Interactive Computer Graphics (2^e édition)
International Student Edition-Mc Graw Hill (1981)

Annexe.

1) Structure physique.

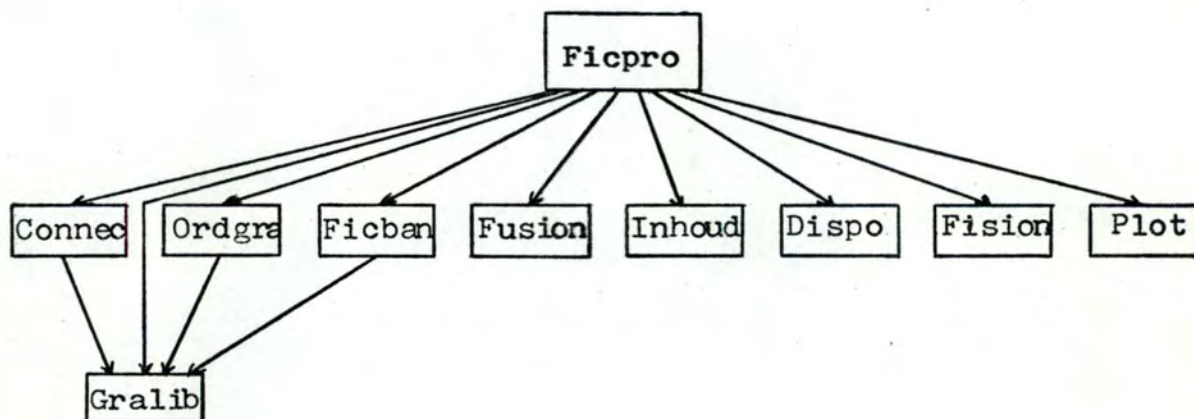


Fig. A.1 : Structure physique du programme FICPRO.

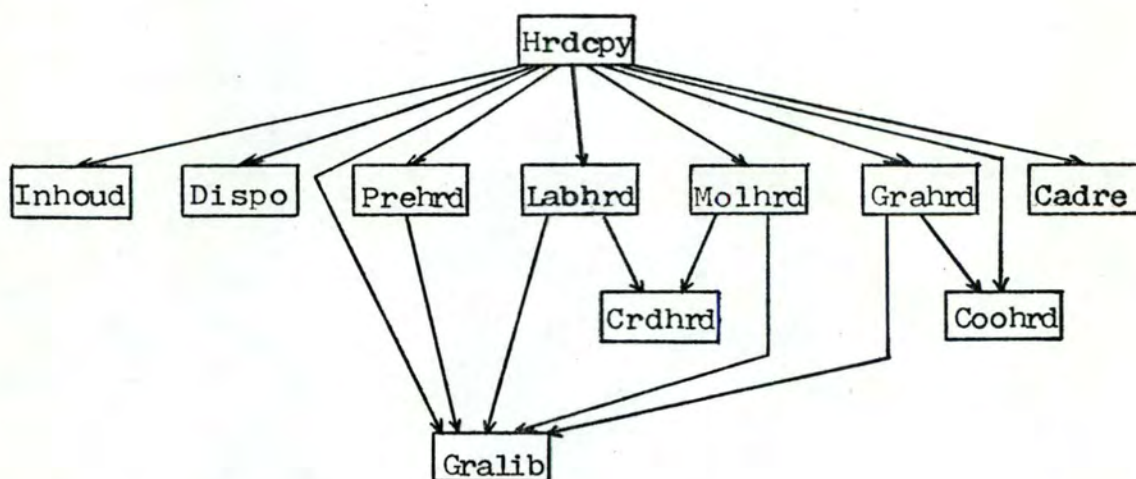


Fig. A.2 : Structure physique du programme HRDCPY.

2) Structure d'overlay.

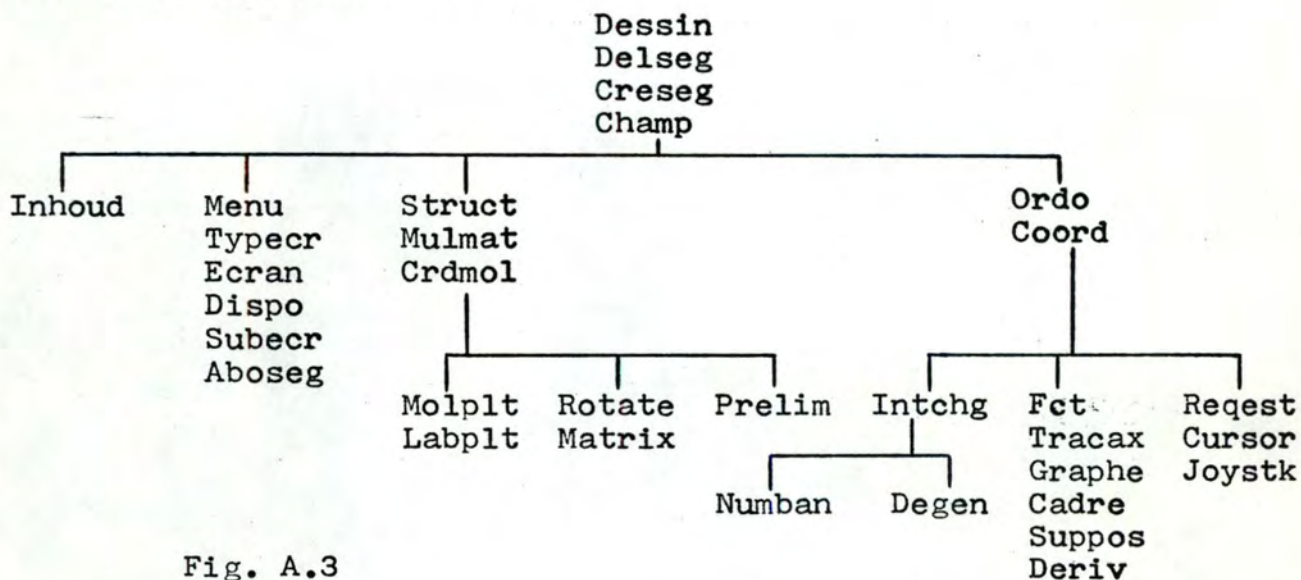
a) Programme Dessin.

Fig. A.3

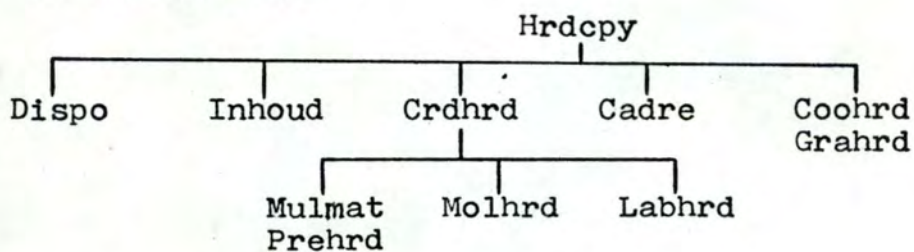
b) Programme HRDCPY.

Fig. A.4

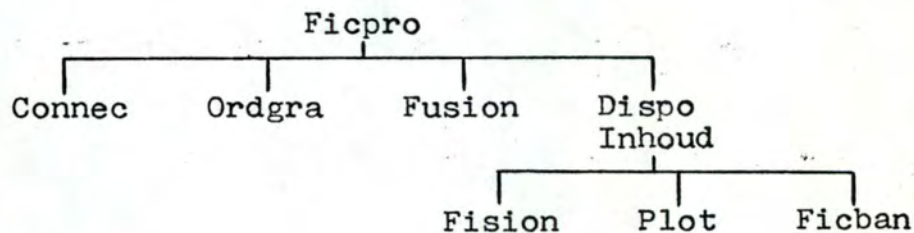
c) Programme FICPRO.

Fig. A.5